Learning a Facial Expression Embedding Disentangled from Identity

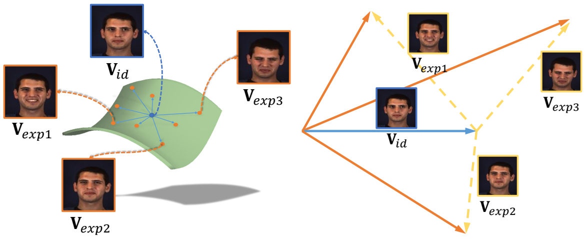
Wei Zhang[1]*, Xianpeng Ji[1]*, Keyu Chen[2]*, Yu Ding[1]*, Changjie Fan[1]
[1]Virtual Human Group, Netease Fuxi AI Lab
[2]University of Science and Technology of China
{zhangwei05,jixianpeng,dingyu01,fanchangjie}@corp.neatease.com
cky95@mail.ustc.edu.cn

Abstract

The facial expression analysis requires a compact and identity-ignored expression representation. In this paper, we model the expression as the deviation from the identity by a subtraction operation, extracting a continuous and identity-invariant expression embedding. We propose a Deviation Learning Network (DLN) with a pseudo-siamese structure to extract the deviation feature vector. To reduce the optimization difficulty caused by additional fully connection layers, DLN directly provides high-order polynomial to nonlinearly project the high-dimensional feature to a low-dimensional manifold. Taking label noise into account, we add a crowd layer to DLN for robust embedding extraction. Also, to achieve a more compact representation, we use hierarchical annotation for data augmentation. We evaluate our facial expression embedding on the FEC validation set. The quantitative results prove that we achieve the state-of-the-art, both in terms of fine-grained and identity-invariant property. We further conduct extensive experiments to show that our expression embedding is of high quality for expression recognition, image retrieval, and face manipulation.

1. Introduction

Facial expression plays a vital role in human social communication. Humans are very skilled at perceiving expressions, which is non-trivial for computers. The development of human-computer interaction requires reasonable facial expression representation. Studies on resolving this problem have been conducted for decades in the expression analysis area. The previous methods, such as *Facial Action Coding System* (FACS) [11] and categorical expression model [7] [28], try to build discrete expression representation with semantic definitions. However, these methods ignore the big variance within the emotional classes and few of them can handle well the identity-entangled problem. Therefore, most of the existing expression representations are still lack of enough capacity for extracting fine-grained and identity-invariant expression information from given faces and somehow block the downstream tasks such as expression recognition [28], image retrieval [36] and face manipulation [39].



(a) Expression manifold (b) Deviation module

Figure 1: Description of expression manifold and our Deviation model.

In this paper, we develop a novel expression embedding framework being capable of learning a continuous and smooth space for expressions from face images. To represent the complicated and subtle expression, we follow the idea [36] of using contrastive comparison to learn the expression distribution. The optimization goal is to map two similar expressions within a triplet close to each other while away from the third one. Besides, based on the perspective that expressions should be lying on a manifold closing to the identity [15] (See Fig. 1a), we model the facial expression as a deviation from the facial identity representation like Fig. 1b. In this way, we disentangle the facial identity attribute from the source face in an explicit manner. To the best of our knowledge, this is the first work that tries to resolve the expression embedding problem by intentionally disentangling the face identity attribute.

Our expression embedding learning framework called Deviation Learning Network (DLN) is composed of three modules (See Fig. 2): a deviation module for extract-

*Equal contribution. Yu Ding is the corresponding author.

ing identity-invariant deviation features, a high-order module for mapping the high-dimensional features to a low-dimensional manifold, and a crowd layer for eliminating the label noise. Specifically, the deviation model has a pseudo-siamese structure, one branch for the original face representation and the other for the identity. The expression deviation feature is calculated by subtracting the identity attribute from face representation. Moreover, instead of using additional neural network layers that cause optimization difficulty due to the gradient shatter [1], we use the high-order polynomial of the deviation for nonlinear mapping in the high-order module. The high-order idea proposed in [3] aims to model inter-layer interactions while ours is for enhancing the fitting performance. Then the crowd layer is used to alleviate different annotators' label bias, ensuring a more robust expression embedding.

The main contribution of this paper lies in three aspects. First, we tackle the challenge of learning an identity-invariant facial expression embedding through an innovative deviation learning network. Secondly, we propose the high-order module to improve the fitting performance from the high-dimensional expression space to a low-dimensional manifold. Third, we enhance the expression embedding, in terms of robustness with the crowd layer and also in terms of fine-grained property with the hierarchical-annotated triplets. Extensive experiments demonstrate the great potential of our method in emotion recognition, image retrieval, and face manipulation tasks.

2. Related work

Expression Embedding FACS [11] proposes to describe an expression as the combination of a set of distinct local Action Units (AUs). However, human perception of facial expression often relies on a full face instead of a local face region. On the other hand, the detection of multiple AUs is still challenging due to positive occurrence and negative competition between AUs [29][42]. Differently, another method represents an expression by learning a low-dimension nonlinear manifold embedded in a face image space [5] [35]. Most of the reported works based on discrete tasks such as expression recognition [7] [37] [28] [16] and ignore the large variance within class. Some of them even can be only used on the aligned faces in the laboratory environment [5] [35]. As a result, the extracted embedding can not reasonably figure out continuous and smooth expression space and reflect a subtle change of expression. FECNet [36] uses a simple feed-forward neural network to extract a continuous expression embedding with the help of annotated triplets. In addition to directly learn an expression manifold, a number of works focuses on expression-related tasks, including expression recognition [28], expression image retrieval [36], and expression manipulation [8]. A well-defined and powerful expression embedding would be helpful to improve the performances in these tasks.

Disentangled Representation Due to the disentangling nature of Generative Adversarial Network (GAN), some works use GANs to extract expression representation [23] [9] [22]. TDGAN [40] proposes a two-branch GAN to learn to disentangle the expression information from other facial attributes. Info-GAN [6] can learn disentangled expression representations by maximizing the mutual information but struggle to train stably. Also, some works [27] [22] [21] use the identity-invariant contrastive losses to minimize the differences between the samples with the same discrete expression category. Koujan et al. [17] proposes a continuous expression regression approach but limited by the 3D morphable model. Most of them are based on a discrete expression task and/or can't be directly employed in the complex in-the-wild environment. Our efforts are made on developing ease of training disentangled expression representation with in-the-wild data.

3. Methodology

This section will present the carefully-designed Deviation Learning Network (DLN). First, we propose a deviation module that enforces a deviation from the identity representation to describe identify-invariant expression (Sec. 3.1). In Sec. 3.2, we introduce the high-order module that learns the complicated nonlinear mapping from the high-dimensional deviation space to a low-dimensional manifold. Furthermore, we improve the robustness of the DLN with a crowd layer design (Sec. 3.3). Finally, we employ a hierarchical annotation strategy to make the learned expression embedding more compact and fine-grained. (Sec. 3.4).

3.1. Deviation Module

Regarding an arbitrary human face as the combination of its identity and expression components, the expression deviation feature is expected to encode the identity-invariant information from the original face. We formulate such an expression deviation learning in a disentangled manner. Let $\mathbf{V}_{face}$ be the feature vector of an input face; $\mathbf{V}_{id}$ the identity attribute, the expression attribute $\mathbf{V}_{exp}$ is given by:

$$\mathbf{V}_{exp} = \mathbf{V}_{face} - \mathbf{V}_{id}. \tag{1}$$

This formulation is supported by early analysis of facial expression manifold [5][11][35]. Given groups of expressions belonging to different individuals, those expressions from the same identity always gather around in a sub-manifold and close to the neutral face (i.e., identity). Besides, the semantic-similar expressions from different identities are analogous on the expression manifold. Based on the observation, we propose to subtract the identity attribute from the $\mathbf{V}_{face}$ to learn the expression from deviation (See Fig. 1b).

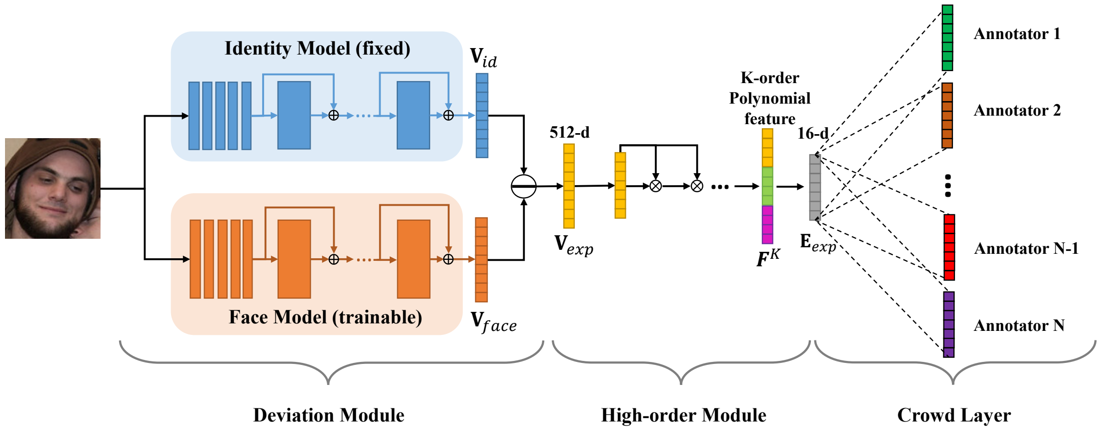


Figure 2: Proposed Deviation Learning Network (DLN) for creating facial expression embedding. We use a pre-trained FaceNet as the Identity Model and fix its parameters. The *face* model has the same structure as the Identity Model but with trainable parameters. To build a low-dimension manifold of expression embedding, we propose a high-order module for better optimization, instead of additional neural network layers. Then a crowd layer is taken to alleviate label noise. In the test, we use expression embedding $\mathbf{E}_{exp}$ only, ignoring the crowd layer.

To this end, we design a deviation module with two pseudo-siamese branches referring to an *identity* model and a *face* model, respectively. As shown in Fig. 2, the *identity* model is responsible for extracting $\mathbf{V}_{id}$ from an input face. Meanwhile, we train the *face* model to learn the feature vector of an input face $\mathbf{V}_{face}$. Note that the *identity* model and the *face* model share the same network structure but with different parameters.

For the *identity* model, we use the pre-trained Inception-Resnet faceNet [34] with the fixed parameters to produce a 512 dimensional vector $\mathbf{V}_{id}$ referring to reliable identity attribute of an input face image. Then we copy the network structure and retrain the *face* model with the initialized parameters same as the pre-trained *identity* model. In this way, the *face* model that benefits from a good initialization will seek for an optimal point around $\mathbf{V}_{id}$ and output another 512 dimensional vector $\mathbf{V}_{face}$. After that, we send the expression deviation vector $\mathbf{V}_{exp}$ to the high-order module.

3.2. High-order Module

To make an expression embedding more compact and effective, we need to fit the deviation $\mathbf{V}_{exp}$ from a high-dimensional (512-dimensional) feature space to a low-dimensional (16-dimensional) manifold. A straightforward method is to utilize several neural layers. Considering many neural layers existing in the *face* model, additional neural layers for reducing the dimension of $\mathbf{V}_{exp}$ may lead to optimization difficulty (e.g. gradient shattering) [1]. From the aspect of universal approximation theorem, the neural layers is actually to fit the features in a high-order space. For the sake of better optimization, we directly provide the high-order polynomial to facilitate the learning of non-linear mapping in the high-order module. The high-order terms can be formulated as:

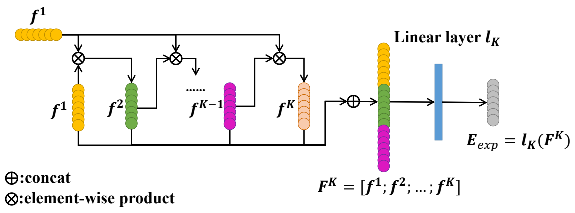


Figure 3: Process of obtaining high-order polynomial feature F^K and achieving the final expression embedding $\mathbf{E}_{exp}$.

$$\begin{aligned} f^k &= f^{k-1} \otimes f^1; \\ F^K &= \left[f^1; f^2; f^3; ...f^K \right] \end{aligned} \tag{2}$$

where $k \in \{1, 2, \cdots K\}$. f^1 is 16-dimensional vector from $\mathbf{V}_{exp}$ through a linear layer. f^k is k-order term of f^1. $\otimes$ refers to element-wise product. F^K is a K-order polynomial feature vector concatenating all the high-order terms from 1 to K. So, F^K is a $16 \times K$ dimensional feature vector. Then, F^K is fed into a linear layer l_k to obtain the final 16 dimensional expression embedding. Fig. 3 describe the detailed process of the high-order module. To alleviate the gradient vanishing and exploding problem caused by high-order polynomial, we perform mean variance normalization for each order [12]. The experiments (described below) show the optimal performance with K=3.

3.3. Crowd Layer

In FEC [36] dataset, there are 43 annotators participating in labelling about 500k triplets and each triplet is distributed to six annotators for *anchor/positive/negative* judgement. In this circumstance, the personal subjectivity needs to be seriously considered since it may bring underlying bias to the final expression embedding. To stabilize our model training process against potential noise data, i.e., inconsistent or wrong labelled triplets, we propose a crowd layer [32] to eliminate the annotator's bias. We use a fully connection layer for each annotator to learn an individual embedding from the common expression embedding, capturing the annotator-specific label bias (See Fig. 2).

To obtain a compact and continuous expression embedding, we follow the idea from FECNet [36] to learn the expression manifold by the triplet loss. Given an image triplet T annotated by K annotators, we denote the annotated results as $\mathrm{T}^{(k)} = \{I_{a_k}, I_{p_k}, I_{n_k}\}$, I_{a_k} (*anchor*) shares more similar expression with I_{p_k} (*positive*) than with I_{n_k} (*negative*), judged by the k-th annotator. The expression embeddings of three face images are denoted as $\{E_{a_k}, E_{p_k}, E_{n_k}\}$. The DLN is constrained to map E_{a_k} close to E_{p_k} while away from E_{n_k}. Therefore, the entire training loss of the DLN is defined by the weighted sum of multiple triplet loss:

$$\mathcal{L}(\mathrm{T}) = \frac{1}{K}\sum_{k=1}^{K} \mathbf{W}_k \mathcal{L}_{tri}(\mathrm{T}^{(k)}); \qquad (3)$$

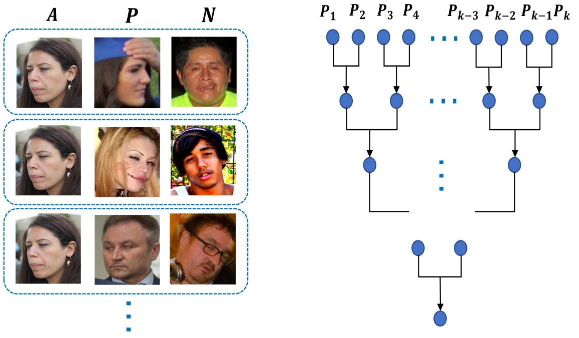
Here, W_k represents the accuracy of the k-th annotator's labeling, reflecting the trustworthiness of the personal judgement. It is calculated by N_k^{agr}/N_k. N_k^{agr} is the number of triplets labelled by k-th annotator that achieve agreement with others, while N_k is the annotation amount of the k-th annotator. During the training stage, we use the output embedding of different annotator-specific layers to compute the triplet loss L_{tri}, as formulated by

$$\begin{aligned}\mathcal{L}_{tri}(\mathrm{T}^{(k)}) = \max(0, \|E_{a_k} - E_{p_k}\|_2^2 - \|E_{a_k} - E_{n_k}\|_2^2 + m) \\ + \max(0, \|E_{a_k} - E_{p_k}\|_2^2 - \|E_{p_k} - E_{n_k}\|_2^2 + m).\end{aligned} \qquad (4)$$

The gradients of the annotator-specific layers will be backpropagated to refine the expression embedding to be more robust. In the inference stage, we drop the annotator-specific layers and directly use the expression embedding before the crowd layer. Since it can be robust on noise data, our model can be extensively trained on the less-confident annotated data, like the weak pairs in the FEC dataset.

3.4. Hierarchical Annotation

Even though we have used all the available manual labelled data including strong and weak pairs (described below) in FEC [36] dataset to train the expression embedding, there are still left information worthy of exploration. Since there are many cases that one anchor image appears in several pairs, we can group the annotated triplets by the anchor images (see Fig. 4a). We have known the similarity relation in a single pair, but the multiple *positive* expressions also have different levels of similarity with the same *anchor*. The comparisons between these positives are even more valuable to the final embedding sub-space, since they provide more fine-grained information to differ images with high similarity.



(a) Group case (b) Hierarchical annotation

Figure 4: Illustration of hierarchical annotation. Fig. 4a shows an example of group case which consists of several triplet pairs with the same anchor. Fig. 4b describes the strategy of Hierarchical Annotation. In each level, we provide one anchor and two positives from different pairs to the annotators. The annotators determine which one is more similar to the anchor expression. The last level will determine which positive is the most like anchor.

Accordingly, we use a simple but effective method called *hierarchical annotation* to annotate these challenging cases. In order to alleviate the cost of manual labor, we adopt the tournament-based design from [41] to arrange the annotation process. As described in Fig. 4b, we first pick k *positive* expression images from the same group w.r.t. all close to the same *anchor* expression. Then we randomly organize these expressions in pairs and ask different annotators to choose one from two *positives* which is more similar to the *anchor*. The chosen ones will be re-organized again and compared with each other in pairs, until there is only one expression winning. During this process, we actually annotated more ”difficult” triplet data. Besides, because of the pyramid structure of the hierarchical annotation design, we can automatically generate more triplet results by chain comparisons. For example, supposed that P_1 wins P_2, P_3 wins P_4 and P_1 wins P_3 (see Fig. 4b), it is reasonable to infer that P_1 also wins P_4.

In practice, we set k=16 and our human annotators only need to compare 8+4+2+1=15 pairs while another four times results would be directly inferred by the *hierarchical annotation* strategy. We maintain the robustness of the human annotation process by joint participation of multiple persons. For hard case, the annotator may choose “uncertain” option and thus the case will be assigned to another one. Furthermore we ensure each group of annotation will be repeated at least three times. With all the new-labelled triplets, we continue to train our DLN to learn the local-scale relationships between similar expressions. Since those *positive* expressions are already mapped closed to each other, fine-tuning on such sensitive data will significantly refine our expression embedding to be more and more compact.

4. Experiments

In this section, we will first introduce our experiment settings including datasets and implementation details. Then we evaluate our method on various metrics. We also perform several ablation studies to prove the effectiveness of each module in our framework. Finally, we conduct multiple applications by using the learned expression embedding to explore its potential ability.

4.1. Datesets

FEC dataset [36] FEC dataset is a large-scale and multi-identity dataset consists of 155,943 faces images, along with 500,203 triplets including *anchor/positive/negative* annotations made by human perception. Among all the triplets, strong pairs refer to those which receive at least two-thirds of raters agreement with each other, and weak pairs receive only half of the raters agreement with each other. There are 357,749 strong pairs and 49,986 weak pairs in the training set. We test our model on the held-out FEC validation set with 41,594 strong pairs. The triplet prediction accuracy is used as a metric for evaluation.

Expression datasets We perform in-the-wild expression recognition on two often-used datasets, AffectNet [28] and RAF-DB [19]. AffectNet is a large in-the-wild expression dataset with 450,000 images categorized into eight basic expressions (neutral, happiness, surprise, sadness, anger, disgust, fear, contempt). To avoid the data imbalance issue, seven basic expressions except for contempt are usually evaluated. RAF-DB consists of face images of seven single expression categories and compound categories. We only choose images with single expression labels for training and testing. RaFD [18] contains images of 67 subjects displaying 8 emotional expressions (Anger, disgust, fear, happiness, sadness, surprise, contempt, and neutral). There are three different gaze directions and five camera angles for each category. CK+ [25] dataset contains 593 video sequences from a total of 123 different subjects. Out of these videos, 327 are labelled with one of seven expression classes. The MMI [30] dataset contains over 2900 videos and images from 75 subjects with annotations of action units and emotions. We use RaFD, CK+, and MMI to measure the identity disentanglement and RaFD is also used to conduct the face manipulation application.

4.2. Implementation details

In experiments, we adopt FaceNet [34] pre-trained on VggFace2 [4] dataset as the *identity* model of DLN and fix its parameters all the time. The *face* model shares the same network structure and parameter initialization with the *identity* model. The entire DLN model is trained on the FEC [36] dataset with a batch size of 30. The input images are resized to 224 × 224. We use the SGD optimizer with a momentum of 0.9 and the learning rate at 2×10^{-4}. The training process is applied on an NVIDIA TitanXP Graphics Card with PyTorch [31]. For more network structure and training details, please refer to the supplementary material.

Table 1: Prediction accuracy (in %) of DLN and previous works on FEC validation set. The best is indicated bold.

Method	Validation set			
	C_1	C_2	C_3	M
AFFNet-CL-P	49.2	59.8	50.4	53.3
FECNet	77.1	85.1	82.6	81.8
DLN(Ours)	**81.8**	**88.3**	**85.6**	**85.4**

$\mathbf{C_1}$: One-class; $\mathbf{C_2}$: Two-class; $\mathbf{C_3}$: Three-class; **M**: Mean.

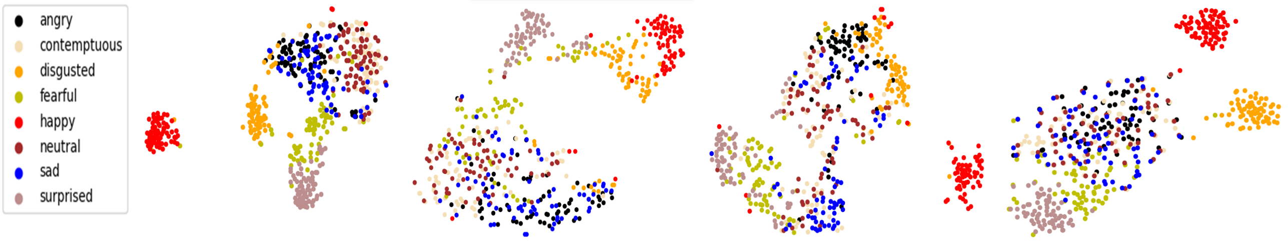
4.3. Evaluation

We evaluate the learned expression embedding by two experiments. First, we apply triplet prediction experiment on the FEC [36] validation set and compare our prediction accuracy with other approaches. Then we quantitatively and qualitatively evaluate the identity-disentanglement property of our embedding and the other competitive expression representation methods.

Triplet prediction We conduct triplet prediction comparisons with FECNet [36] and AffectNet [28]. In the held-out validation set of FEC [36] dataset, there are three kinds of triplet data (one-class, two-class and three-class), each triplet is specified with ground-truth annotation. After going through the expression embedding module, we compute the distances between every two faces within a triplet and determines the *anchor/positive/negative* object, i.e., predicted annotation. By calculating the matching percentage of the predicted and the ground-truth annotation, we can give the triplet prediction accuracy as shown in Tab. 1.

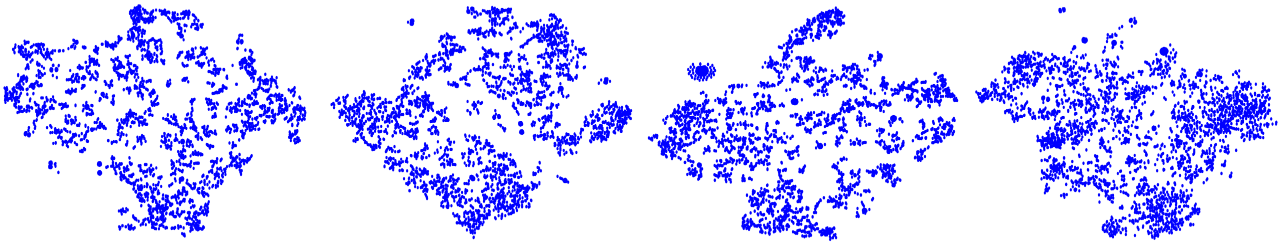
In comparison, FECNet [36] employs a network that connects fixed FaceNet and Densenet directly and AFFNet-CL-P [36] represents the penultimate layer of FECNet continuously trained on AffectNet [28]. From the quantitative results, we find that our method achieves better prediction accuracy, either in each class or in total. Particularly, in the one-class triplet prediction test, which is the most challenging case because the input expressions are very similar to each other, we reach 81.8% of accuracy, 4.7% higher than the FECNet. This proves our model better represent fine-grained expressions.

Identity disentanglement A good expression embedding is supposed to be identity-ignored, which means that semantic-similar expressions of different individuals should be closely-embedded on the expression manifold. So, we compare the identity disentanglement property of different expression representations including DLN (ours), FEC-



(a) DLN (*Ave-Var*=0.0089) (b) FECNet [36] (*Ave-Var*=0.011) (c) AFF [36] (*Ave-Var*=0.039) (d) 3DMM [2] (*Ave-Var*=0.0097)

Figure 5: 2D t-SNE visualization of expression embedding with different methods, based on the RaFD [18] dataset. It is observed that embeddings from our DLN are more compact in each expression class. *Ave-Var* is the average of embedding variances of all the expression classes.



(a) C^a: DLN^1 (*Acc*=0.837) (b) C^b: DLN^1+CL (*Acc*=0.841) (c) C^c: DLN^1+HA (*Acc*=0.842) (d) C^d: DLN^3 (*Acc*=0.846)

Figure 6: 2D t-SNE visualization of expression embedding from four conditions (C) including DLN^1 without Crowd Layer (CL) and Hierarchical Annotation (HA) (Fig. 6a), DLN^1 with CL but without HA (Fig. 6b), DLN^1 without CL but with HA (Fig. 6c), and DLN^3 without CL and HA (Fig. 6d). The displayed points refers to 5,000 images randomly sampled from the FEC training set. *Acc* represents the prediction accuracy on the FEC validation set.

Table 2: The average variance of each emotion class on RaFD, CK+, and MMI datasets. The best is indicated bold.

Dataset	FECNet	AFFNet	3DMM	Ours
CK+	0.011	0.021	0.019	**0.010**
MMI	0.021	0.057	0.019	**0.017**
RaFD	0.011	0.039	0.0097	**0.0089**

Net [36], AFFNet-CL-P [28] and 3DMM [2]. We use the RaFD[18], CK+ [25] and MMI [30] datasets and average the variance of each emotion class's embedding. Also, due to the paper size, we only show the t-SNE [26] visualization of embedding distribution of each method on RaFD dataset(See Fig. 5). The average variance can be seen in Tab. 2.

From the qualitative and quantitative results, it can be observed that the embedding with the same expression gathers closer to each other on our expression manifold and the distribution variance is smaller, in comparison to the other three methods. It indicates that our DLN performs well on disentangling expression from identity and generates independent expression embedding space.

4.4. Ablation Study

We perform ablation studies to demonstrate the effectiveness of the high-order module, crowd layer, and hierarchical annotation in our DLN. By replacing each module with different settings, we compute the triplet prediction accuracy of each solution. The results are shown in Tab. 3.

High-order module. To investigate the impact of the high-order module, we change the value of order k increasingly from 1 to 5. Specifically, k at 1 refers to the common-used neural network layers with no high-order terms; k at other values refers to the high-order module with corresponding terms from 1 to k-order. Accordingly, DLN^k refers to DLN with the high-order terms from 1 to k. Results in Tab. 3 indicate that the high-order module with k from 2 to 4 outperforms the neural network layers ($k = 1$). This confirms the effectiveness of the high-order module. The order of 3 helps to get the best triplet prediction accuracy above all. In other experiments, DLN employs the order of 3. The reason of performance decreasing at $k = 4$ and 5 could be the overfitting issue since they bring much more parameters than the low order cases.

Crowd layer. Tab. 3 shows the experimental results with and without the crowd layer component. As observed, the

Table 3: Prediction accuracy (in %) of ablation experiments on the FEC validation set. The best order is indicated by brackets. The best is indicated in bold.

Method			Validation set			
Order	CL	HA	C$_1$	C$_2$	C$_3$	M
DLN^1	×	×	79.3	86.9	84.3	83.7
DLN^2	×	×	80.4	87.7	84.8	84.4
DLN^3	×	×	[80.8]	[87.8]	[85.1]	[84.6]
DLN^4	×	×	79.9	87.6	84.5	84.1
DLN^5	×	×	79.7	87.2	83.5	83.5
DLN^3	√	×	81.5	88.0	85.5	85.1
DLN^3	×	√	81.7	87.9	85.6	85.2
DLN^3	√	√	**81.8**	**88.3**	**85.6**	**85.4**

C$_1$: One-class; **C$_2$**: Two-class; **C$_3$**: Three-class; **M**: Mean; **CL**: Crowd Layer; **HA**: Hierarchical Annotation. ×: no use; √: use;

crowd layer increases the accuracy from 84.6% to 85.1%. It suggests that the crowd layer being capable of eliminating the annotator's subjective bias and data noise is indeed effective in learning a more powerful expression embedding sub-space.

Hierarchical annotation. In Tab. 3, we also compare the experimental results generated with and without the hierarchical annotation module. By adding the hierarchically annotated triplets to training data, the final prediction accuracy is improved from 84.6% to 85.2%. It demonstrates the effectiveness of our hierarchical annotation strategy and confirms that the hierarchical annotation does provide more fine-grained information and our framework has the capability to capture the local-scale expression difference.

Additionally, we also plot the 2D t-SNE [26] under separate condition of crowd layer, hierarchical annotation and high-order module (See Fig. 6). Comparing C^a to C^b, C^c, or C^d in Fig. 6, we can observe that the crowd layer (C^a), the hierarchical annotation (C^b) and the high-order module C^d make the expression embedding more compact. This observation is consistent with their effectiveness reported in the above.

4.5. Applications

The expression embedding from our method can be applied to other expression-related tasks, such as expression recognition, expression image retrieval and face manipulation. We will demonstrate the potential application with our expression embedding.

4.5.1 Expression Recognition

Our extracted embedding can be taken as features directly to classify expression combined with K-Nearest Neighbor classifier (KNN). The first two lines in Tab. 4 show the results with 200 neighbours in KNN. It is observed that our embedding from DLN outperforms FECNet without re-training. The comparison result can be attribute to that our method extracts more compact and precise expression embedding. On the other hand, DLN can be applied to expression recognition when re-training on the expression dataset. The experiment show that DLN is closed to the state-of-the-art methods on several datasets [28] [19]. Notice that, to achieve the good performance, various complicated tricks are employed in the state-of-the-art methods. Both RAN [38] and gACNN [20] employed attention mechanisms to highlight the importance of facial regions for expression recognition and alleviate the pose and occlusion problems; PAENet [14] and CPG [13] based on continuous learning use several datasets like VGGFace2 [4], IMDb-Wiki [33], FotW [10] and AffectNet [28], while our DLN only makes use of FaceNet based on VGGFace2 [28]. Our expression embedding achieve the comparable results without any additional operations. This confirms the effectiveness of our model on discriminating expression categories.

Table 4: Accuracy (in %) of DLN and previous works on facial expression recognition.The best and second are indicated using bold and brackets alone, respectively.

Method	AffectNet	RAF-DB
FECNet+KNN[36]	29.4	59.7
DLN+KNN(ours)	34.2	65.0
RAN[38]	59.5	**86.9**
PAENet[14]	**65.3**	-
CPG[13]	63.6	-
gACNN[20]	58.8	85.1
FECNet[36]	58.9	73.0
DLN(Ours)	[63.7]	[86.4]

4.5.2 Expression Image Retrieval

Another practical application of expression embedding is expression image retrieval. Like [36], we use the nearest neighbor search method in expression embedding space to address this task. We construct a query set with 25 images and use CelebA [24] as the database. We retrieve nearest N (N=1,2,...,10) images from the database by computing the distances within the expression embedding space of ours and the FECNet [36]. We conduct a user study by asking ten participants to choose the more similar expression images retrieved by our model and the FECNet. If the given results are too similar to be judged, the participant may choose the option ”Uncertain” as well. In Fig. 8, we show the Top-5 statistical analysis of collected votes. The quantitative results indicate that our method obtains more preferring votes than the FECNet, both in each case and in total.

Fig. 7 gives some retrieval samples of FECNet and our DLN. While the images retrieved with FECNet tend to have similar identity information, DLN results contain more various identities. This supports our identity-invariant expres-

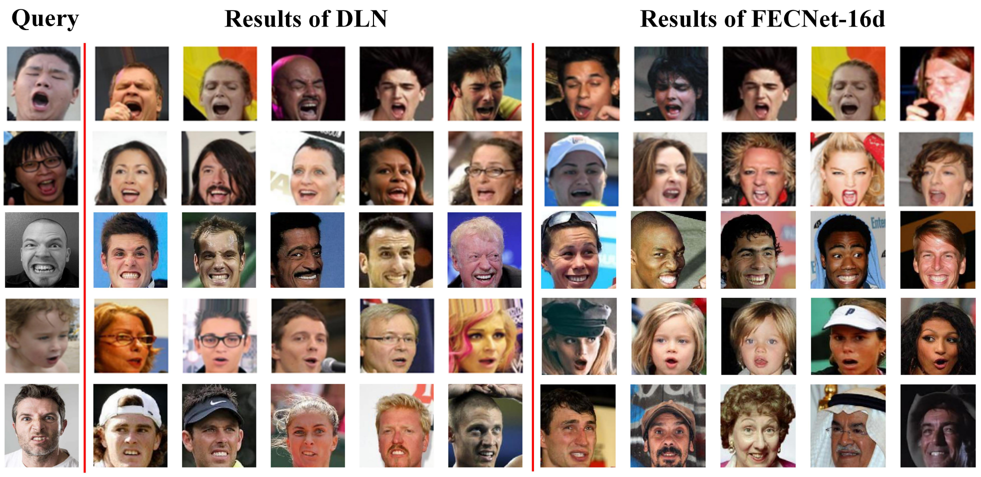


Figure 7: Comparison of image retrieval between DLN (ours) and FECNet-16d. We show the top-5 images retrieved using embeddings. Our results perform better especially on fine-grained expressions. Please zoom in for more details.

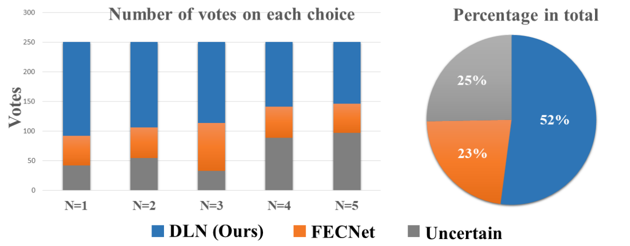


Figure 8: Voting on image retrieval results. Our method gains more preference from the raters, either in each case or in total.

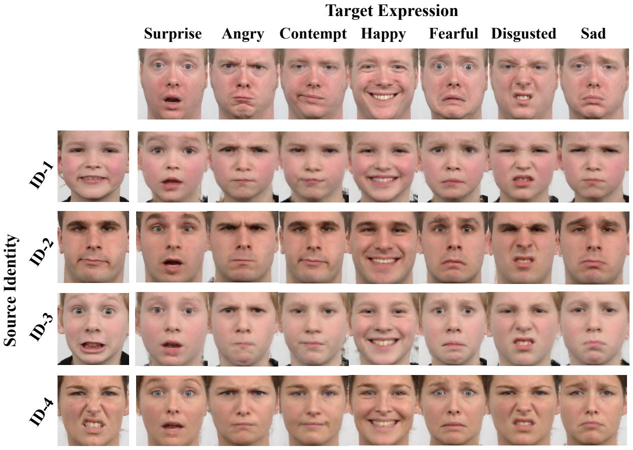


Figure 9: Face manipulation results of RaFD dataset using expression embeddings from DLN. The first column is the input face, the first row is the target expression. The middle part shows the generated faces manipulated with the expression embeddings. Please zoom in for more details.

sion embedding. On the other hand, our DLN can perform better especially on some complex and fine-grained expressions. For example, In the last row of Fig. 7, the query image shows a man is gnashing the teeth and staring in anger or disgust. The results of DLN mainly are related to negative expression like disgust, but the results of FECNet seem like have error expression category.

4.5.3 Face Manipulation

Our DLN expression embedding can be directly used to manipulate human portraits. First, we produce expression embeddings from our DLN and fix them. Then we feed an arbitrary face image of source identity and target expression embedding into a Conditional GAN[39] to generate an expression-manipulated face image. During the training, we use the same identities' supervised data of RaFD [18] that contains corresponding different expressions of each person. While in testing, we generate faces manipulated by another person's expression. Fig. 9 shows the generated faces. It is observed that our generated faces are consistent with the target expressions and source identities, which means that our expression embedding is of high quality in capturing the detailed expression information and disentangling the identity attribute.

5. Conclusion

We have presented a Deviation Learning Network to learn a compact and identity-invariant facial expression embedding by explicitly disentangling the identity attribute. We have demonstrated the effectiveness of the proposed method via an ablation study and extensive quantitative and qualitative experiments. Applications like expression recognition, expression image retrieval, and face manipulation have shown powerful capability and great potential of our learned expression embedding. In the near future, we will refine expression embedding by moving out head pose and taking into account more data in the wild.

References

[1] David Balduzzi, Marcus Frean, Lennox Leary, JP Lewis, Kurt Wan-Duo Ma, and Brian McWilliams. The shattered gradients problem: If resnets are the answer, then what is the question? *arXiv preprint arXiv:1702.08591*, 2017. 2, 3

[2] Volker Blanz and Thomas Vetter. A morphable model for the synthesis of 3d faces. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 187–194, 1999. 6

[3] Sijia Cai, Wangmeng Zuo, and Lei Zhang. Higher-order integration of hierarchical convolutional activations for fine-grained visual categorization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 511–520, 2017. 2

[4] Qiong Cao, Li Shen, Weidi Xie, Omkar M Parkhi, and Andrew Zisserman. Vggface2: A dataset for recognising faces across pose and age. In *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)*, pages 67–74. IEEE, 2018. 5, 7

[5] Ya Chang, Changbo Hu, Rogerio Feris, and Matthew Turk. Manifold based analysis of facial expression. *Image and Vision Computing*, 24(6):605–614, 2006. 2

[6] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *Advances in neural information processing systems*, pages 2172–2180, 2016. 2

[7] Yuedong Chen, Jianfeng Wang, Shikai Chen, Zhongchao Shi, and Jianfei Cai. Facial motion prior networks for facial expression recognition. In *2019 IEEE Visual Communications and Image Processing (VCIP)*, pages 1–4. IEEE, 2019. 1, 2

[8] Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8789–8797, 2018. 2

[9] Chris Donahue, Zachary C Lipton, Akshay Balsubramani, and Julian McAuley. Semantically decomposing the latent spaces of generative adversarial networks. *arXiv preprint arXiv:1705.07904*, 2017. 2

[10] Sergio Escalera, Mercedes Torres Torres, Brais Martinez, Xavier Baró, Hugo Jair Escalante, Isabelle Guyon, Georgios Tzimiropoulos, Ciprian Corneou, Marc Oliu, Mohammad Ali Bagheri, et al. Chalearn looking at people and faces of the world: Face analysis workshop and challenge 2016. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 1–8, 2016. 7

[11] E Friesen and Paul Ekman. Facial action coding system: a technique for the measurement of facial movement. *Palo Alto*, 3, 1978. 1, 2

[12] Mohit Goyal, Rajan Goyal, and Brejesh Lall. Learning activation functions: A new paradigm of understanding neural networks. *arXiv preprint arXiv:1906.09529*, 2019. 3

[13] Ching-Yi Hung, Cheng-Hao Tu, Cheng-En Wu, Chien-Hung Chen, Yi-Ming Chan, and Chu-Song Chen. Compacting, picking and growing for unforgetting continual learning. In *Advances in Neural Information Processing Systems*, pages 13669–13679, 2019. 7

[14] Steven CY Hung, Jia-Hong Lee, Timmy ST Wan, Chein-Hung Chen, Yi-Ming Chan, and Chu-Song Chen. Increasingly packing multiple facial-informatics modules in a unified deep-learning model via lifelong learning. In *Proceedings of the 2019 on International Conference on Multimedia Retrieval*, pages 339–343, 2019. 7

[15] Zi-Hang Jiang, Qianyi Wu, Keyu Chen, and Juyong Zhang. Disentangled representation learning for 3d face shape. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 11957–11966, 2019. 1

[16] Corentin Kervadec, Valentin Vielzeuf, Stéphane Pateux, Alexis Lechervy, and Frédéric Jurie. Cake: Compact and accurate k-dimensional representation of emotion. *arXiv preprint arXiv:1807.11215*, 2018. 2

[17] Mohammad Rami Koujan, Luma Alharbawee, Giorgos Giannakakis, Nicolas Pugeault, and Anastasios Roussos. Real-time facial expression recognition" in the wild"by disentangling 3d expression from identity. *arXiv preprint arXiv:2005.05509*, 2020. 2

[18] Oliver Langner, Ron Dotsch, Gijsbert Bijlstra, Daniel HJ Wigboldus, Skyler T Hawk, and AD Van Knippenberg. Presentation and validation of the radboud faces database. *Cognition and emotion*, 24(8):1377–1388, 2010. 5, 6, 8

[19] Shan Li and Weihong Deng. Reliable crowdsourcing and deep locality-preserving learning for unconstrained facial expression recognition. *IEEE Transactions on Image Processing*, 28(1):356–370, 2019. 5, 7

[20] Yong Li, Jiabei Zeng, Shiguang Shan, and Xilin Chen. Occlusion aware facial expression recognition using cnn with attention mechanism. *IEEE Transactions on Image Processing*, 28(5):2439–2450, 2018. 7

[21] Xiaofeng Liu, BVK Vijaya Kumar, Ping Jia, and Jane You. Hard negative generation for identity-disentangled facial expression recognition. *Pattern Recognition*, 88:1–12, 2019. 2

[22] Xiaofeng Liu, BVK Vijaya Kumar, Jane You, and Ping Jia. Adaptive deep metric learning for identity-aware facial expression recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 20–29, 2017. 2

[23] Yu Liu, Fangyin Wei, Jing Shao, Lu Sheng, Junjie Yan, and Xiaogang Wang. Exploring disentangled feature representation beyond face identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2080–2089, 2018. 2

[24] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of the IEEE international conference on computer vision*, pages 3730–3738, 2015. 7

[25] Patrick Lucey, Jeffrey F Cohn, Takeo Kanade, Jason Saragih, Zara Ambadar, and Iain 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*, pages 94–101. IEEE, 2010. 5, 6

[26] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008. 6, 7

[27] Zibo Meng, Ping Liu, Jie Cai, Shizhong Han, and Yan Tong. Identity-aware convolutional neural network for facial expression recognition. In *2017 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017)*, pages 558–565. IEEE, 2017. 2

[28] Ali Mollahosseini, Behzad Hasani, and Mohammad H Mahoor. Affectnet: A database for facial expression, valence, and arousal computing in the wild. *IEEE Transactions on Affective Computing*, 10(1):18–31, 2017. 1, 2, 5, 6, 7

[29] Xuesong Niu, Hu Han, Songfan Yang, Yan Huang, and Shiguang Shan. Local relationship learning with person-specific shape regularization for facial action unit detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 11917–11926, 2019. 2

[30] Maja Pantic, Michel Valstar, Ron Rademaker, and Ludo Maat. Web-based database for facial expression analysis. In *2005 IEEE international conference on multimedia and Expo*, pages 5–pp. IEEE, 2005. 5, 6

[31] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. dAlché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. 5

[32] Filipe Rodrigues and Francisco Pereira. Deep learning from crowds. *arXiv preprint arXiv:1709.01779*, 2017. 4

[33] Rasmus Rothe, Radu Timofte, and Luc Van Gool. Dex: Deep expectation of apparent age from a single image. In *IEEE International Conference on Computer Vision Workshops (ICCVW)*, December 2015. 7

[34] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823, 2015. 3, 5

[35] Caifeng Shan, Shaogang Gong, and Peter W McOwan. Appearance manifold of facial expression. In *International Workshop on Human-Computer Interaction*, pages 221–230. Springer, 2005. 2

[36] Raviteja Vemulapalli and Aseem Agarwala. A compact embedding for facial expression similarity. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5683–5692, 2019. 1, 2, 3, 4, 5, 6, 7

[37] Kai Wang, Xiaojiang Peng, Jianfei Yang, Debin Meng, and Yu Qiao. Region attention networks for pose and occlusion robust facial expression recognition. *IEEE Transactions on Image Processing*, 29:4057–4069, 2020. 2

[38] Kai Wang, Xiaojiang Peng, Jianfei Yang, Debin Meng, and Yu Qiao. Region attention networks for pose and occlusion robust facial expression recognition. *IEEE Transactions on Image Processing*, 29:4057–4069, 2020. 7

[39] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8798–8807, 2018. 1, 8

[40] Siyue Xie, Haifeng Hu, and Yizhen Chen. Facial expression recognition with two-branch disentangled generative adversarial network. *IEEE Transactions on Circuits and Systems for Video Technology*, 2020. 2

[41] Juyong Zhang, Keyu Chen, and Jianmin Zheng. Facial expression retargeting from human to avatar made easy. *IEEE Transactions on Visualization and Computer Graphics*, 2020. 4

[42] Kaili Zhao, Wen-Sheng Chu, Fernando De la Torre, Jeffrey F Cohn, and Honggang Zhang. Joint patch and multi-label learning for facial action unit detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2207–2216, 2015. 2