



Article

A New Joint Training Method for Facial Expression Recognition with Inconsistently Annotated and Imbalanced Data

Tao Chen ¹, Dong Zhang ^{1,*}  and Dah-Jye Lee ² 

¹ School of Electronics and Information Technology, Sun Yat-sen University, Guangzhou 510006, China; chent229@mail2.sysu.edu.cn

² Department of Electrical and Computer Engineering, Brigham Young University, Provo, UT 84602, USA; djlee@byu.edu

* Correspondence: zhangd@mail.sysu.edu.cn

Abstract: Facial expression recognition (FER) plays a crucial role in various applications, including human–computer interaction and affective computing. However, the joint training of an FER network with multiple datasets is a promising strategy to enhance its performance. Nevertheless, widespread annotation inconsistencies and class imbalances among FER datasets pose significant challenges to this approach. This paper proposes a new multi-dataset joint training method, Sample Selection and Paired Augmentation Joint Training (SSPA-JT), to address these challenges. SSPA-JT models annotation inconsistency as a label noise problem and selects clean samples from auxiliary datasets to expand the overall dataset size while maintaining consistent annotation standards. Additionally, a dynamic matching algorithm is developed to pair clean samples of the tail class with noisy samples, which enriches the tail classes with diverse background information. Experimental results demonstrate that SSPA-JT achieved superior or comparable performance compared with the existing methods by addressing both annotation inconsistencies and class imbalance during multi-dataset joint training. It achieved state-of-the-art performance on RAF-DB and CAER-S datasets with accuracies of 92.44% and 98.22%, respectively, reflecting improvements of 0.2% and 3.65% over existing methods.

Keywords: facial expression recognition; joint training; annotation inconsistency; class imbalanced; learning with noisy label; long-tailed learning



Citation: Chen, T.; Zhang, D.; Lee, D.-J. A New Joint Training Method for Facial Expression Recognition with Inconsistently Annotated and Imbalanced Data. *Electronics* **2024**, *13*, 3891. <https://doi.org/10.3390/electronics13193891>

Academic Editor: George A. Tsihrintzis

Received: 8 September 2024

Revised: 27 September 2024

Accepted: 30 September 2024

Published: 1 October 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Facial expression recognition (FER) is a critical research topic in computer vision and affective computing, as it has a wide range of applications in human–computer interaction [1], sentiment analysis [2], security monitoring [3], healthcare [4,5], and other practical implementations. Depending on the scenarios of data collection, FER datasets can be categorized into lab-controlled FER datasets [6–8] and in-the-wild FER datasets [9–14]. Rather than acquiring samples from controlled environment, in-the-wild FER datasets collect facial expression samples in natural, unconstrained environments. Compared to lab-controlled FER datasets, in-the-wild datasets typically include more samples and present more complex scenarios, which provide diversified training data for FER tasks.

In recent years, deep neural networks (DNNs)-based FER methods have obtained remarkable success on many in-the-wild datasets. In general, the performance of a DNN is dependent on the amount of training data, and increasing the size of the training set is usually an effective way to improve model performance [15]. Combining the training sets from different datasets focused on the same task is a straightforward approach to enlarge the training set. However, due to the diversity of facial expressions and the subjective nature of their interpretation, along with factors such as image quality and annotator discrepancies, inconsistent annotation standards (a.k.a. annotation inconsistency) often exist among different in-the-wild FER datasets [16]. Additionally, these datasets typically exhibit an imbalanced class distribution with a long tail [17], where a few classes (known

as head classes) contain the majority of the data, while most classes (known as tail classes) have very few samples. This imbalance arises because certain facial expressions naturally occur more frequently in real life [11]. Affected by these two factors, directly combining these in-the-wild FER datasets leads to a significant change of distribution between the combined training set and the original target validation set, which may negatively impact the performance of the network [18].

To address the problem of annotation inconsistency among FER datasets, researchers have proposed various approaches. Zeng et al. proposed to improve the annotation consistency by combining human-annotated labels with network-predicted labels [16]. Wang et al. employed a self-attention mechanism to evaluate the annotation quality and applied a regularized ranking and label correction strategy to progressively standardize the annotations within the training set, reducing labeling discrepancies [19]. Yu et al. [18] utilized continuous expression labels and aligned the discrete labels across different FER datasets with techniques of subset selection, continuous label mapping, and discrete label re-annotation. These methods alleviated the annotation inconsistencies among FER datasets and improved FER performance through dataset joint training.

However, a significant limitation of existing methods is that they do not account for the impact of class imbalance on the effectiveness of joint training. Our exploratory experiments found that increasing the numbers of samples from different classes may exert a distinct influence in the training of an FER network; e.g., increasing the number of tail class samples has a more substantial effect on improving model performance than increasing the number of head class samples. Figure 1 shows the class distribution in the training sets of two commonly used FER datasets, RAF-DB [9,10] and AffectNet [11] (7-class). As illustrated in Figure 1, the samples of fear, anger, disgust, and surprise are only a very small proportion in both datasets, collectively accounting for less than one fourth of the total samples. In this paper, we consider these four expression classes as tail classes, while the remaining three expressions—sadness, neutrality, and happiness—are treated as head classes.

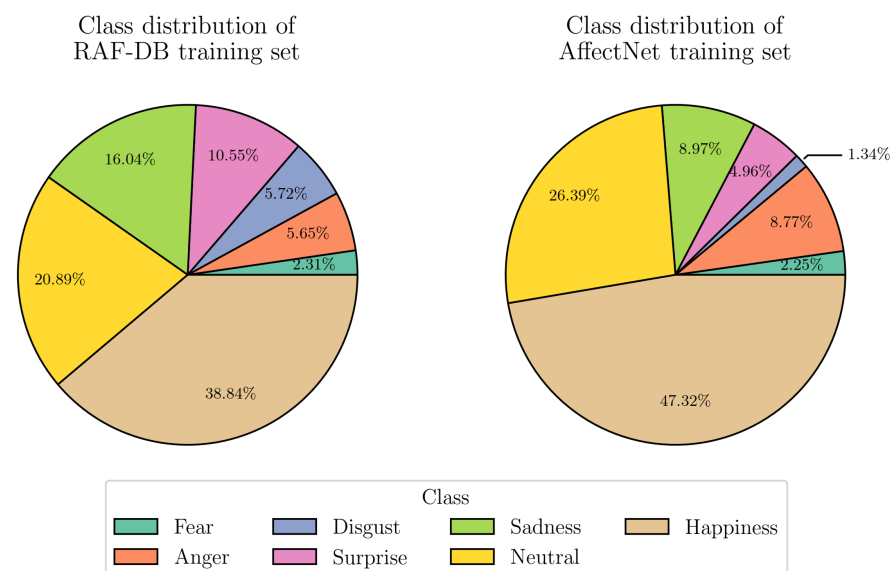


Figure 1. Class distribution of RAF-DB and AffectNet training sets.

To better understand the challenges posed by class imbalance, we conduct experiments using the two datasets mentioned above. We designate RAF-DB as the target dataset, which serves as the primary dataset for training and evaluation. AffectNet is utilized as the auxiliary dataset, providing additional samples to augment the training data. Figure 2 illustrates the impact of augmenting different class samples on model performance evaluated on the test set of RAF-DB. When the model is trained solely on the training set of RAF-DB,

the recognition accuracy on the test set of RAF-DB is 86.25%. However, when the training sets of RAF-DB and AffectNet are directly combined for training, the recognition accuracy decreases by 1.63% due to annotation inconsistencies between the two datasets. Merging the training set of RAF-DB with only the samples in the head class and tail class from the training set of AffectNet, the recognition accuracy decreases by 1.15% and increases by 0.21%, respectively. These results indicate that enlarging the target dataset with tail class samples from the auxiliary dataset improves the model performance on the validation set of the target dataset. Conversely, directly combining the target and auxiliary datasets or augmenting only the head class samples may harm the model performance, as it does not take account of the class imbalance. Given the prevalent issue of class imbalance in FER datasets, where head and tail classes are generally consistent across datasets, neglecting this imbalance may significantly undermine the effectiveness of dataset joint training. Therefore, addressing both annotation inconsistency and class imbalance remains a critical challenge for improving the effectiveness of dataset joint training in FER tasks.

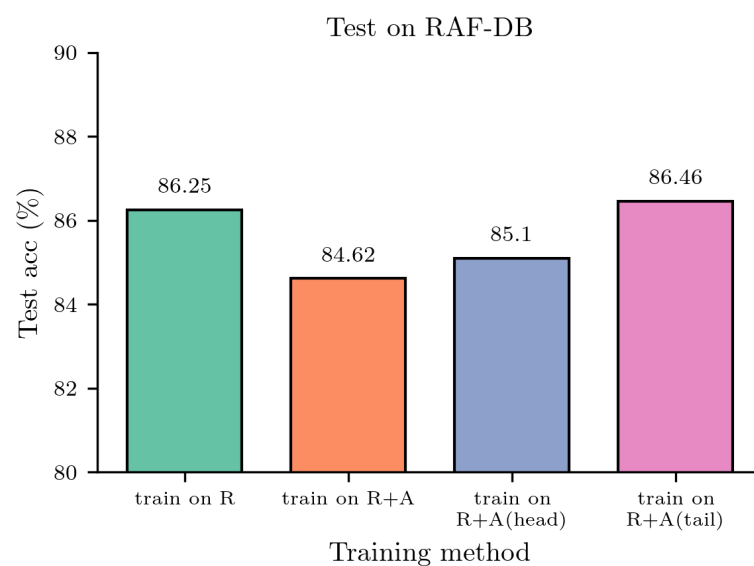


Figure 2. The performance of classification on the RAF-DB test set with different training sets. *A* denotes a training set of AffectNet while *R* denotes a training set of RAF-DB. *head* denotes head classes (sadness, neutral and happiness), *tail* denotes tail classes (fear, anger, disgust and surprise).

In this paper, we propose a new joint training method for FER multi-dataset joint training. To tackle the problem of annotation inconsistency among different FER datasets, we model it as a label noise problem. We assume that the labels in the training set of the target dataset are accurate or clean, whereas the auxiliary dataset may contain noisy labels due to varied annotation standards. Inspired by label noise handling techniques [20], we implement a sample selection process to identify clean samples from the auxiliary dataset as they are consistent with the annotation standard of the target dataset. These selected samples are then directly merged with the training set of the target dataset to expand the overall dataset size while maintaining annotation consistency. For the problem of class imbalance, we recognize that the noisy samples from the auxiliary dataset, although not being selected due to annotation inconsistency, still contain valuable information related to facial expressions. We argue that noise is not always detrimental if used appropriately; it can enhance the robustness of model and mitigate class bias. To address class imbalance, we propose a dynamic matching algorithm that pairs the clean samples from tail classes with noisy samples from the auxiliary dataset. Data augmentation is then performed on these matched sample pairs to create new tail class samples with additional background noise. This approach not only leverages the auxiliary dataset more effectively but also

alleviates the class bias by enriching the tail classes with diverse background information, improving the overall accuracy of joint training.

Our contributions are summarized as follows:

(1) We provide a detailed analysis of how class distribution imbalance across different datasets impacts the performance of joint training in FER, which is supported by empirical evidence and experimental analysis.

(2) We propose a comprehensive framework, Sample Selection and Paired Augmentation Joint Training (SSPA-JT), that simultaneously addresses annotation inconsistency and class imbalance. This framework introduces an innovative data augmentation method by pairing annotation-inconsistent samples from the auxiliary dataset with tail class samples from the target dataset, enriching the diversity of tail class samples and improving model performance.

(3) We demonstrate through extensive experiments that the proposed SSPA-JT framework achieves better or comparable performance compared to existing FER methods, including other joint training approaches.

2. Related Work

Early FER methods primarily relied on handcrafted descriptors to extract facial features, such as Local Binary Patterns (LBPs) [21], Histograms of Oriented Gradients (HOGs) [22], and Scale-Invariant Feature Transform (SIFT) [23]. However, these feature engineering-based methods suffer from limited robustness as the extracted features are susceptible to variations in image quality, lighting, angle, and occlusion. With the rise of deep learning, FER methods based on deep learning have gradually become the mainstream approach. Compared to traditional methods, deep learning methods significantly enhance recognition performance by automatically learning features. Training on large-scale datasets enables them to capture the diversity and complexity of the data and improves the generalization capability of the model. However, deep learning methods still face several challenges, particularly those related to label noise and class imbalance inherent in FER datasets.

2.1. Learning with Noisy Labels

Due to the subjectivity of manual labeling, errors during data collection, and various other reasons, some samples are often annotated with noisy labels—labels that do not align with the latent true values. Noisy labels may lead models to learn incorrect patterns and reduce the generalization ability of models. This challenge has been widely studied in recent years, leading to the development of a specialized subfield known as Learning with Noisy Labels (LNL). Among the various approaches within LNL, sample selection has emerged as an important and effective technique.

Sample selection methods are typically based on the small-loss assumption, which posits that neural networks tend to fit clean samples before overfitting to noisy labels [24]. The core idea of sample selection is to identify and either remove or reweight samples with noisy labels, reducing their impact on model training. Common approaches include confidence-based sample selection [20,25], loss-based sample selection [26,27], and model consistency-based sample selection. For instance, the co-teaching method trains two neural networks that mutually filter each other's samples to avoid the influence of noisy labels [28]. Li et al. introduced a method that dynamically sets thresholds based on the memory strength of each instance during training, allowing the selection of reliable instances [20]. These instances are then divided into different subsets, with distinct regularization strategies applied to each, effectively enhancing the robustness of the model to noisy labels.

In addition to sample selection, there are other strategies to address noisy labels. For example, label correction methods improve label quality by leveraging model confidence scores or auxiliary information from related tasks to correct potentially noisy labels [29,30]. Loss function adjustment methods, on the other hand, reduce the sensitivity of models to

noisy labels by modifying traditional loss functions, mitigating the negative impact of label noise [31–33]. Some approaches combine these strategies to further enhance the robustness of the model against label noise.

Traditional active learning (AL) techniques, such as uncertainty-based sample selection [19,34], differ from threshold-based sample selection approaches. While threshold-based selection methods can be seen as a “hard” selection strategy that either accepts or rejects samples based on predefined rules, uncertainty-based AL adopts a “soft” selection strategy by assigning lower weights to uncertain samples, reducing their influence during training to avoid overfitting. The weights in AL are typically derived from the model itself, making the method more sensitive to the current state of learning of the model. As a result, the effectiveness of uncertainty-based sample selection can be heavily influenced by the training dynamics of the model, in contrast to threshold-based methods, which are less dependent on the internal biases of the model.

Due to the subjective nature of facial expression annotation and the complexity of expression variations, the issue of noisy labels in FER is particularly severe, making it a significant research challenge in this field. Many methods have been developed specifically to address noisy labels in FER. For example, Zhang et al. [34] quantified the uncertainty of samples by comparing their relative difficulty and assigned lower weights to uncertain samples to reduce the impact of noisy labels. Wang et al. [35] employed emotion ambiguity-sensitive learning and negative correlation diversity enhancement strategies to filter clean and ambiguous samples from noisy datasets, promoting diverse feature representations to enhance the robustness of the model against label noise. Zhang et al. [36] proposed a method that randomly erases input images and utilizes flip attention consistency during training to suppress the focus of the model on noisy labels, preventing the model from memorizing noisy samples. Wu et al. [37] leveraged facial landmark information to mitigate the effects of noisy labels. By incorporating key modules for label distribution estimation and expression-landmark contrastive loss, they improved the quality of supervision under noisy label conditions and enhanced the robustness of the expression feature extractor.

2.2. Long-Tailed Learning

Long-tailed distribution is another critical challenge in machine learning. In many scenarios, class distributions are long-tailed, meaning that a few classes dominate the majority of samples, while many classes have noticeably fewer samples. This imbalance of classes usually negatively impacts model performance, leading to reduced recognition ability for the tail classes. To address the challenges of long-tailed distribution, researchers have proposed various long-tailed learning methods.

Data resampling is a common approach for addressing long-tailed distributions. This technique involves either undersampling the head classes or oversampling the tail classes to reduce class imbalance [38,39]. For instance, Kang et al. [38] achieved high performance in long-tailed recognition by employing simple instance-balanced sampling to learn high-quality representations and adjusting only the classifier. Similarly, Hu et al. [39] effectively improved tail class recognition by dividing the long-tailed dataset into balanced subsets and applying an incremental learning strategy alongside a class-incremental few-shot learning paradigm to handle class imbalance and few-shot learning challenges in large-scale long-tailed datasets.

Improving the loss function is another important approach to address long-tailed problems. Classic loss functions like Focal Loss [40] and Class-Balanced Loss [41] assign different weights to samples from different classes, enabling the model to focus more on tail classes during training.

Additionally, to enrich the information of minority classes, specific data augmentation techniques can be used to generate additional samples [42,43]. For example, Park et al. [44] improved the generalization capacity of classifier in long-tailed classification by enhancing the diversity of tail class samples using the rich contextual backgrounds of head class samples.

In the field of FER, long-tailed distribution is also a critical challenge. Due to the varying natural occurrence rates of different expressions, FER datasets often exhibit a long-tailed distribution. Gao et al. [45] addressed the issue of long-tailed distribution in large-scale FER datasets by constructing multiple data subsets and applying a progressive pruning technique, resulting in a model that performs well on imbalanced datasets. Similarly, Zhang et al. [46] improved the recognition performance of minority classes without compromising the performance on majority classes by extracting additional knowledge related to minority classes from both major and minor class samples and using rebalanced attention maps and label smoothing to guide the model.

The aforementioned methods demonstrate that enhancing the robustness of a model against label noise often improves its performance, particularly in tasks where label noise is significant, such as in FER datasets. However, current label noise learning techniques mostly focus on training with a single FER dataset. In scenarios involving multi-dataset joint training, label errors arising from inconsistent annotation standards can also be modeled as label noise issues. Therefore, label noise learning methods are promising for addressing inconsistencies in dataset annotations. Additionally, merging FER datasets with imbalanced class distributions may exacerbate the imbalance. Overcoming the impact of class distribution imbalance during joint training remains a critical challenge. To address these issues, we propose the SSPA-JT framework, which performs joint training across multiple FER datasets with inconsistent annotations. The proposed framework integrates sample selection-based label noise robustness techniques with data augmentation-based class imbalance learning methods. This approach aims to alleviate the impact from annotation inconsistency and class imbalance in joint dataset training.

3. Method

The performance of joint training with multiple FER datasets suffers from inconsistent annotation standards and class imbalance. To address these challenges, we propose a new joint training framework, SSPA-JT, to alleviate the influence from annotation inconsistency and class imbalance and improve the effectiveness of joint training for FER.

First, we model annotation inconsistency as a label noise problem by treating samples with annotation bias in the auxiliary dataset as mislabeled samples. To expand the training dataset while maintaining annotation consistency, inspired by sample selection methods in the Learning with Noisy Labels (LNL), we filter out clean samples that align with the annotation standard of the target dataset from the auxiliary dataset and merge them into the target dataset.

Next, to alleviate the negative influence of class imbalance in FER datasets, we enhance the background diversity of tail class samples. We design a dynamic matching algorithm that pairs noisy-labeled samples from the noisy set with samples of tail classes in the clean set and generates new training samples through data augmentation. It fully leverages not only all available data but also improves the robustness of the model against background variations, and it reduces the influence of class bias as well.

We use \mathcal{D}_T to denote the target dataset and \mathcal{D}_A the auxiliary dataset. The entire dataset \mathcal{D} is the union of \mathcal{D}_T and \mathcal{D}_A . The clean set and the noisy set obtained by sample selection at epoch t are denoted as \mathcal{D}_C^t and \mathcal{D}_N^t . The minibatch sampled from \mathcal{D} is denoted as \mathcal{B} . We denote (x, y) as a sample of \mathcal{D} , where x represents the image and y represents the label of this sample. The feature vector extracted from x by the feature extractor is denoted as f . The prediction of x is denoted as $p = (p^1, \dots, p^C)$, where C is the number of classes.

The overview of SSPA-SJ is shown in Figure 3. The training of SSPA-SJ consists of two stages: Sample Selection (SS) and Paired Augmentation (PA). The model is composed of a feature extractor module, Mixture of Experts (MoE) module, Sample Selection Module (SSM), and Paired Augmentation Module (PAM). In our implementation, we utilize two different backbones for feature extraction: ResNet-18 and ARM. The feature extractor and MoE are used in both stages, while the SSM and PAM are specific to the SS stage and PA

stage, respectively. Additionally, L_{CE} in the figure refers to the cross-entropy loss used for training the model.

In the SS stage, the MoE provides k predictions for each sample. The SSM assigns a set of dynamic thresholds to each sample in \mathcal{D} with each threshold corresponding to a specific prediction made by the MoE, and these thresholds change in each training epoch to determine whether a sample is clean. Based on this threshold and the predictions of MoE, the SSM classifies samples into the clean set or noisy set in each epoch. Only the clean samples from the clean set are used for model parameter optimization. At the end of the SS stage, a clean sample set \mathcal{D}_C , comprising samples from both the target and a portion of the auxiliary datasets, and a noise sample set \mathcal{D}_N , comprising the remaining auxiliary dataset samples, are created. These two sets are kept constant and used for training in the PA stage.

In the PA stage, each minibatch \mathcal{B} , randomly sampled from \mathcal{D} and used for training, contains both clean and noisy samples. The PAM pairs each noisy sample in \mathcal{B} with a clean sample using a designed dynamic matching algorithm, and then it replaces the noisy sample with new samples through data augmentation to form a new minibatch $\tilde{\mathcal{B}}$. Owing to the designed dynamic paired algorithm and data augmentation, $\tilde{\mathcal{B}}$ is more balanced in class distribution.

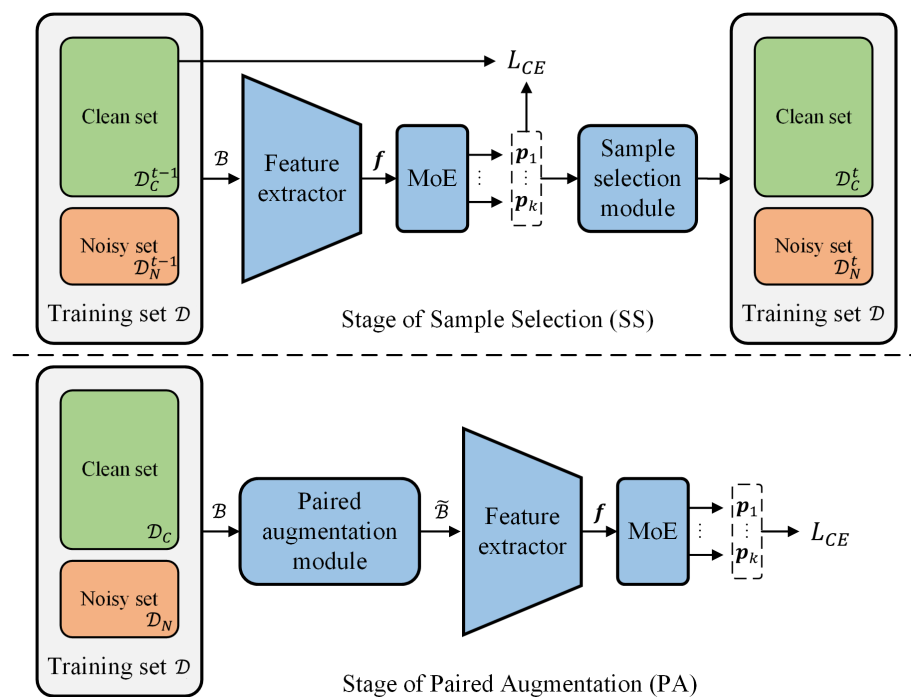


Figure 3. Overview of our proposed SSPA-JT joint training framework.

3.1. Mixture of Experts Module

Inspired by recent advancements in large language models (LLMs) and ensemble learning techniques in LNL, we designed a Mixture of Experts (MoE) module to enhance our framework. The MoE module leverages multiple specialized sub-networks, or experts, to increase the model’s capacity and provide a broader perspective for subsequent sample selection. As illustrated in Figure 4, the MoE module consists of a gating mechanism and a set of feedforward networks (FFNs), where each FFN is implemented as a single-layer fully connected network that acts as an individual expert.

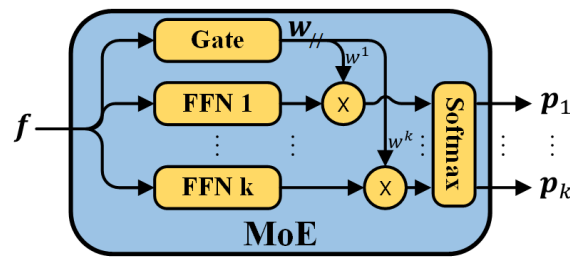


Figure 4. The structure of the MoE.

Given an input feature vector f , the MoE module first processes this input with the gating mechanism. The gate determines the relevance of each expert to the given input by computing a set of importance weights. Specifically, the gate assigns a weight to each of the k experts, indicating how much each expert contributes to the final output. The outputs of these experts will be multiplied by their corresponding computed weights.

Each expert processes the input independently. The output of each expert is scaled by the associated weight from the gate. The weighted outputs from all experts then pass through a Softmax function, which produces the final predictions, indicating the classification for that sample.

$$w = \text{Gate}(f) = (w^1, \dots, w^k) \tag{1}$$

$$p_i = \text{Softmax}(\text{FFN}_i(f) \cdot w_i), \forall i \in [1, \dots, k] \tag{2}$$

where w denotes the weight vector, p_i denotes the prediction of the i -th FFN, and k denotes the number of experts.

By dynamically adjusting the weights of different experts for each input, the MoE module allows the model to adapt to various input characteristics and improves its flexibility and overall performance in addressing the diverse challenges inherent in FER tasks. Additionally, this module generates multiple predictions for each sample and provides the subsequent sample selection modules with a broader perspective to assess the cleanliness of the sample based on these varied predictions.

3.2. Sample Selection Module

Annotation inconsistencies among different datasets are a key factor that hinders the effectiveness of joint training. Samples with annotation biases in the auxiliary dataset can be treated as noisy label samples. To enhance annotation consistency in the combined dataset, we design the SSM to select samples from the auxiliary dataset that are consistent with the annotation standard of the target dataset (a.k.a. clean samples).

Research in the field of LNL found that DNNs tend to learn simple samples before difficult ones, and their confidence on a sample reflects how well they have learned from that sample. Since noisy labels are harder for DNNs to learn, we can filter out potential noisy samples based on the confidence levels of DNNs [20].

SSM maintains a buffer that records k thresholds for each sample, where k corresponds to the number of experts in the MoE described in Section 3.1. These thresholds are dynamically updated during training using an Exponential Moving Average (EMA) method:

$$\tau_i^t = \eta \tau_i^{t-1} + (1 - \eta) \text{Max}(p_i^t), \forall i \in [1, \dots, k], \tag{3}$$

where τ_i^t is the threshold value recorded by SSM for the i -th expert at the t -th epoch, η denotes the EMA momentum coefficient, and $\text{Max}(p_i^t)$ is the maximum value of the prediction probability vector for the i -th expert at epoch t , indicating the confidence of model on that sample. The EMA update method ensures the stability of these threshold values.

After updating the threshold values in each epoch, SSM compares the k predictions from the MoE with the corresponding k thresholds for each sample (x, y) :

$$F_C = [p_i^t(y) > \tau_i^t] \wedge \cdots \wedge [p_k^t(y) > \tau_k^t], \quad (4)$$

where $F_C \in \{\text{True}, \text{False}\}$, \wedge denotes the logical AND operator. $p_i^t(y)$ is the predicted probability of the i -th expert at epoch t on class y . A sample is considered clean if the predicted probability for its label y exceeds the corresponding threshold for all experts. Otherwise, it is deemed a noisy sample. By leveraging the diversity of MoE, SSM gains a more comprehensive view of each sample, which aids in retaining more reliable samples.

Algorithm 1 describes the implementation of SSM in detail.

Algorithm 1 Sample Selection Module

Require: Predictions p_1, \dots, p_k of MoE for sample (x, y) ; thresholds $\tau_1^{t-1}, \dots, \tau_k^{t-1}$ of (x, y) ; clean set \mathcal{D}_C^t ; noisy set \mathcal{D}_N^t .

Ensure: Updated clean set \mathcal{D}_C^t ; updated noisy set \mathcal{D}_N^t .

- 1: **for** each i in $1, \dots, k$ **do**
 - 2: Update τ_i^{t-1} using Formula (3) to obtain τ_i^t ;
 - 3: **end for**
 - 4: Using Formula (4) to obtain F_C
 - 5: $\mathcal{D}_C^t = \mathcal{D}_C^t \cup \{(x, y)\}$ if F_C is True else $\mathcal{D}_N^t = \mathcal{D}_N^t \cup \{(x, y)\}$
 - 6: **return** $\mathcal{D}_C^t, \mathcal{D}_N^t$
-

3.3. Paired Augmentation Module

Considering that class imbalance within FER datasets usually negatively affects the performance of joint training, we propose the Paired Augmentation Module, which closely collaborates with the SSM, to enhance the balance of class distribution and maximize the utilization of all available training samples. Our idea is to increase the number of samples in tail classes by enriching the background diversity.

As mentioned in Section 3.2, the SSM module divides the dataset \mathcal{D} into two complementary subsets, i.e., the clean set \mathcal{D}_C and the noisy set \mathcal{D}_N . Consequently, a randomly sampled minibatch \mathcal{B} from \mathcal{D} is also composed of two complementary subsets: \mathcal{B}_C , consisting of clean samples, and \mathcal{B}_N , consisting of noisy samples. We denote the number of samples in these subsets as N_C and N_N , respectively.

First, we count the number of samples in each class within \mathcal{B}_C and form a vector $\mathbf{h} \in \mathbb{N}^C$ that represents the class distribution in \mathcal{B}_C , where C is the number of classes, and h_i denotes the number of samples in the i -th class. Next, we perform random sampling with replacement from \mathcal{B}_C to obtain N_N samples, which are then randomly paired with each sample in \mathcal{B}_N . We use \mathbf{s} to denote the class distribution of these sampled samples, which are analogous to \mathbf{h} . In this way, we obtain N_N sample pairs, each consisting of a sampled clean sample (x_S, y_S) from \mathcal{B}_C and a noisy sample (x_N, y_N) from \mathcal{D}_N .

Next, we apply the data augmentation method of Mixup [47] on each sample pair to generate a new sample \tilde{x} with the label \tilde{y} inherited from the clean sample:

$$\lambda \sim \text{Beta}(\alpha, \beta) \quad (5)$$

$$\tilde{x} = \lambda x_S + (1 - \lambda)x_N \quad (6)$$

$$\tilde{y} = y_S \quad (7)$$

where α and β are hyperparameters to control the Beta distribution, and λ is a value sampled from this distribution. This step enriches the background diversity of the clean samples by blending them with noisy samples while preserving the clean sample labels. Finally, we replace the noisy samples x_N in \mathcal{B} with the newly generated samples \tilde{x} to form a new minibatch $\tilde{\mathcal{B}}$. The batch size of $\tilde{\mathcal{B}}$ is identical to \mathcal{B} , with a class distribution of $\mathbf{h} + \mathbf{s}$, and all samples in $\tilde{\mathcal{B}}$ participate in the model training.

However, the above steps are still not sufficient to balance the class distribution. To increase the background diversity of tail class samples and enrich the tail class data, the classes with fewer samples in \mathcal{B}_C should be sampled more frequently and combined with noisy samples. Specifically, we aim to minimize the variance of the class distribution vector $\mathbf{h} + \mathbf{s}$ in $\tilde{\mathcal{B}}$, which can be formulated as a constrained optimization problem:

$$\begin{aligned} \min \quad & \sigma(\mathbf{h} + \mathbf{s}) \\ \text{s.t.} \quad & \sum_{i=1}^C s_i = N_N \\ & 0 \leq s_i \leq \begin{cases} 0, & \text{if } h_i = 0 \\ N_N, & \text{otherwise} \end{cases}, \forall i \in [1, \dots, k] \end{aligned} \quad (8)$$

In formula (8), s_i is 0 when $h_i = 0$, as it is impossible to sample from \mathcal{B}_C for class i . In other cases, s_i can range from 0 to N_N , as sampling from \mathcal{B}_C is performed with replacement, and the sum of all elements in \mathbf{s} should equal the number of elements in \mathcal{B}_N , i.e., N_N . We solve this optimization problem using the Sequential Least Squares Programming (SLSQP) algorithm [48], and the obtained solution is discretized to yield the final class distribution vector \mathbf{s} to guide the sampling from \mathcal{B}_C .

Algorithm 2 describes the implementation of PAM in detail.

Algorithm 2 Paired Augmentation Module

Input: Minibatch \mathcal{B} with a clean subset \mathcal{B}_C and a noisy subset \mathcal{B}_N .

Output: Balanced minibatch $\tilde{\mathcal{B}}$.

- 1: Obtain class distribution vector \mathbf{h} for \mathcal{B}_C by category count.
 - 2: Solve Equation (8) using the SLSQP algorithm and discretize its result to obtain \mathbf{s} .
 - 3: Randomly sample N_N clean samples with replacement from \mathcal{B}_C and ensure that their class distribution satisfies \mathbf{s} .
 - 4: Pair the sampled clean samples with the noisy samples in \mathcal{B}_N , obtaining N_N sample pairs.
 - 5: **for** each pair $((x_S, y_S), (x_N, y_N))$ **do**
 - 6: Generate a new sample \tilde{x} using Mixup according to Equations (5) and (6).
 - 7: Set the label of \tilde{x} to $\tilde{y} = y_S$.
 - 8: Replace (x_N, y_N) in \mathcal{B}_N with the newly generated samples (\tilde{x}, \tilde{y}) .
 - 9: **end for**
 - 10: $\tilde{\mathcal{B}} = \mathcal{B}_C \cup \mathcal{B}_N$
-

3.4. Joint Training

We present the complete training process of SSPA-JT in Algorithm 3. Before training begins, the target dataset \mathcal{D}_T and the auxiliary dataset \mathcal{D}_A are merged into a single dataset \mathcal{D} . Initially, all samples are assumed to be clean, so the clean set is defined as $\mathcal{D}_C^{-1} = \mathcal{D}_T \cup \mathcal{D}_A$, while the noisy set \mathcal{D}_N^{-1} is empty, allowing the model to warm up on the entire dataset. Throughout the training process, the samples from \mathcal{D}_T are always treated as clean, since they are annotated according to the annotation standard of the target dataset, which ensures consistent annotations. Additionally, all thresholds in the SSM buffer are initialized to $\frac{1}{C}$.

SSPA-JT training is composed of two stages: the Sample Selection (SS) stage and the Pairing Augmentation (PA) stage. During each epoch of the SS stage, the model performs a forward pass on all samples in \mathcal{D} , calculating k prediction probability vectors for each sample. Then, based on the clean and noisy sets determined by the SSM in the previous epoch, the loss is computed only for the samples in the clean set, which is followed by backpropagation to update the model parameters. The SSM also updates the thresholds in its buffer based on the k predictions for each sample in the current epoch and then reclassifies the samples in \mathcal{D}_A , selecting those with consistent annotations as new clean samples.

After the SS stage, a fixed clean set \mathcal{D}_C and a fixed noisy set \mathcal{D}_N are obtained, which are complementary and will be used for training in the following PA stage. In this stage, the PAM is employed to perform paired augmentation on each minibatch \mathcal{B} sampled from \mathcal{D} , yielding a class-balanced $\tilde{\mathcal{B}}$. The classification loss for all samples in $\tilde{\mathcal{B}}$ is calculated to update the model parameters.

Throughout the training, all loss computations are based on the cross-entropy loss function.

Algorithm 3 Joint Training

Input: Target dataset \mathcal{D}_T ; auxiliary dataset \mathcal{D}_A ; sample selection epoch T_S ; max epoch T_{max} .

Output: Trained model $M(\cdot)$ with a feature extractor denoted $E(\cdot)$ and a mixture of expert module denoted $MoE(\cdot)$.

- 1: Initialize $\mathcal{D}_C^{-1} = \mathcal{D}_T \cup \mathcal{D}_A$, $\mathcal{D}_N^{-1} = \emptyset$; Fix \mathcal{D}_T in \mathcal{D}_C ; $\mathcal{D} = \mathcal{D}_T \cup \mathcal{D}_A$;
- 2: Initialize threshold $\tau_1^{-1}, \dots, \tau_k^{-1}$ for each sample (x, y) ;
- 3: Initialize E with pre-training parameters, initialize MoE with random values;
- 4: **for** $t = 0$ in T_{max} **do**
- 5: **if** $t < T_S$ **then**
- 6: $\mathcal{D}_C^t = \emptyset$, $\mathcal{D}_N^t = \emptyset$;
- 7: **for** \mathcal{B} in \mathcal{D} **do**
- 8: **for** (x, y) in \mathcal{B} **do**
- 9: $p_1, \dots, p_k = MoE(E(x))$
- 10: **if** $(x, y) \in \mathcal{D}_C^{t-1}$ **then**
- 11: $\mathcal{L} = \frac{1}{k} \sum_{i=1}^k \mathcal{L}_{CE}(p_i, y)$
- 12: Update parameters of M by minimizing \mathcal{L}
- 13: **end if**
- 14: Update \mathcal{D}_C^t and \mathcal{D}_N^t by Algorithm 1
- 15: **end for**
- 16: **end for**
- 17: **else**
- 18: **for** \mathcal{B} in \mathcal{D} **do**
- 19: Get $\tilde{\mathcal{B}}$ by Algorithm 2
- 20: **for** (\tilde{x}, \tilde{y}) in $\tilde{\mathcal{B}}$ **do**
- 21: $p_1, \dots, p_k = MoE(E(\tilde{x}))$
- 22: $\mathcal{L} = \frac{1}{k} \sum_{i=1}^k \mathcal{L}_{CE}(p_i, \tilde{y})$
- 23: Update parameters of M by minimizing \mathcal{L}
- 24: **end for**
- 25: **end for**
- 26: **end if**
- 27: **end for**

4. Experiments

4.1. Experiment Settings

4.1.1. Datasets

In this study, we evaluated our proposed method using three widely recognized in-the-wild FER datasets: RAF-DB, CAER-S, and AffectNet.

- **RAF-DB:** The RAF-DB [9,10] dataset consists of 30,000 facial images annotated with either basic or compound expressions. Following previous works, we only use the images with seven basic expression categories, utilizing 12,271 training samples and 3068 test samples. The annotations in RAF-DB were created through a combination of 40 human coders and crowdsourcing techniques, providing a diverse and reliable set of expression labels.
- **CAER-S:** Derived from the CAER [12] dataset, the CAER-S subset includes 65,983 images. This dataset is split into a training set with 44,996 images and a test set with

20,987 images. Each image in CAER-S is labeled with one of seven basic expressions: neutral, happiness, sadness, surprise, fear, disgust, and anger.

- **AffectNet:** AffectNet [11] is among the largest FER datasets, containing over one million images obtained from the Internet using 1250 emotion-related keywords. Out of these, 450,000 images have been manually annotated with 11 discrete labels. For our experiments, we concentrate on the seven basic expression categories, consistent with those in RAF-DB and CAER-S, utilizing a total of 283,721 samples corresponding to these seven expressions.

Figure 5 presents some sample images from the these three datasets.

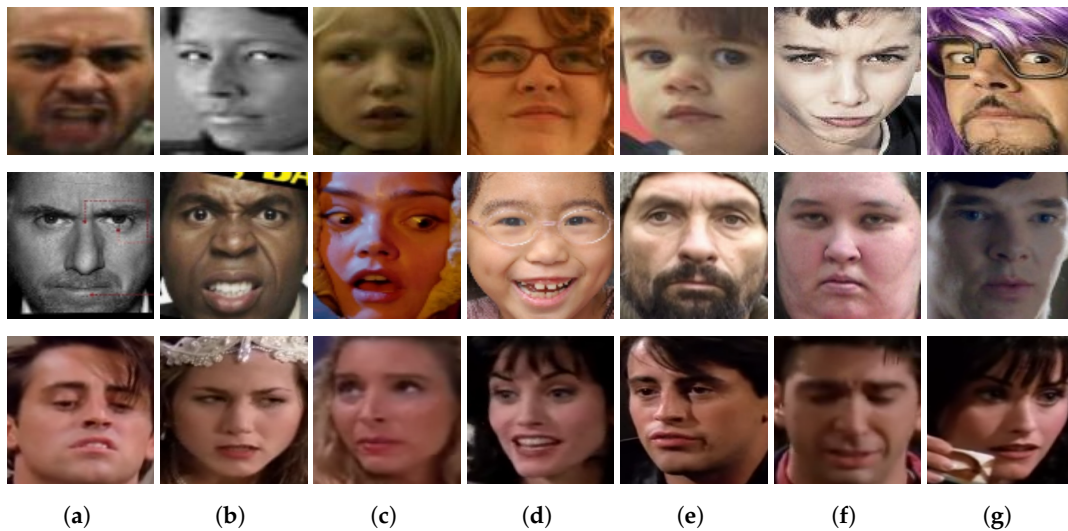


Figure 5. Facial expression samples from three datasets: the first row corresponds to the RAF-DB dataset, the second row to the AffectNet dataset, and the third row to the CAER-S dataset. The columns represent the seven emotion categories: (a) Anger, (b) Disgust, (c) Fear, (d) Happiness, (e) Neutral, (f) Sadness, and (g) Surprise.

4.1.2. Evaluation Protocol and Metrics

In this section, we introduce the evaluation protocol and metrics used to assess the performance of our proposed framework. The joint training algorithm is designed to train on the combined training sets of both the target dataset and the auxiliary dataset. However, validation is performed exclusively on the validation (or test) set of the target dataset. This approach is essential because it ensures that the evaluation remains consistent by using a fixed validation dataset, allowing us to accurately measure whether the expanded training set leads to performance improvements on the target data.

We utilize four evaluation metrics to measure the effectiveness of our method: Accuracy, Macro-Precision, Macro-Recall, and Macro-F1 Score. These metrics are chosen to provide a comprehensive evaluation, particularly in terms of addressing the challenges posed by class imbalance, which often moves the model bias toward majority classes.

- **Accuracy** is the proportion of correctly classified samples to the total number of samples:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (9)$$

where TP stands for the number of true positives, TN stands for the number of true negatives, FP stands for the number of false positives, and FN stands for the number of false negatives.

- **Macro-Precision** evaluates the precision across all classes and then averages them, giving equal weight to each class. It is calculated by

$$\text{Macro-Precision} = \frac{1}{C} \sum_{i=1}^C \frac{TP_i}{TP_i + FP_i} \quad (10)$$

where C represents the number of classes, and TP_i and FP_i are the numbers of true positives and false positives for each class i , respectively. Macro-Precision helps with understanding how well the model can avoid false positives across all classes, particularly the tail classes.

- **Macro-Recall** calculates the recall for each class independently and then takes the average. This metric is crucial for understanding how well the model can identify all instances of each class, including the minority ones:

$$\text{Macro-Recall} = \frac{1}{C} \sum_{i=1}^C \frac{TP_i}{TP_i + FN_i} \quad (11)$$

where FN_i stands for the false negatives for each class i .

- **Macro-F1 Score** is the harmonic mean of Macro-Precision and Macro-Recall, providing a balanced measure of the performance of models across all classes:

$$\text{Macro-F1 Score} = \frac{1}{C} \sum_{i=1}^C \frac{2 \times \text{Precision}_i \times \text{Recall}_i}{\text{Precision}_i + \text{Recall}_i} \quad (12)$$

where Precision_i and Recall_i are the precision and recall for each class i , respectively.

The aforementioned metrics evaluate not only the overall accuracy of the model but also the capability of the model to mitigate the class bias caused by class imbalance. The use of Macro-Precision, Macro-Recall, and Macro-F1 Score ensures that the performance is measured fairly across all classes and gives particular attention to how well the model handles the minority classes. In contrast, Macro-Accuracy may give a misleading impression of performance, as the true positives for minority classes may be very limited, while true negatives can be large, resulting in artificially inflated accuracy that does not accurately reflect the model's predictive capability for those categories. Therefore, we did not choose Macro-Accuracy as a metric.

4.1.3. Details of Implementation

In our experiments, we used RAF-DB or CAER-S as the target datasets and AffectNet as the auxiliary dataset. For the RAF-DB dataset, we used aligned images with seven basic discrete labels, which were resized to 224×224 pixels. For the CAER-S dataset, we detected and aligned all faces using similarity transformation and then resized them to 224×224 pixels. For the AffectNet dataset, we cropped the face images using the provided bounding boxes, applied similarity transformation for alignment, and resized them to 224×224 pixels. We employed random cropping and random horizontal flipping as data augmentation. We used ResNet-18 [49] pretrained on MS-Celeb1M [50] and ARM [51] pretrained on ImageNet [52] as the backbones.

During training, we set the batch size to 128 and used the SGD optimizer with an initial learning rate of 0.01, momentum of 0.9, and weight decay of 0.005. These values were chosen based on preliminary experiments, which indicated that they provide a good balance between convergence speed and stability. We trained the models for 80 epochs on all datasets, employing a cosine annealing learning rate scheduler [53] to gradually decrease the learning rate from the initial value to zero, which has been shown to improve training efficiency. We set the number of sample selection epochs to 40, as this duration effectively balances exploration and exploitation during the training process. The number of experts (k) in the Mixture of Experts (MoE) was set to 3, which allows for sufficient diversity in expert predictions without introducing excessive complexity. The EMA momentum coefficient in the Sample Selection Module (SSM) was set to 0.95, facilitating stable updates of sample selection while retaining important historical information. Finally, the beta distribution

parameters α and β in the Paired Augmentation Module (PAM) were chosen as 5 and 1, respectively, based on empirical results that indicated these values enhance the variability of the sampled data while maintaining a strong representation of the underlying distribution. All experiments were implemented on NVIDIA TITAN Xp GPU and PyTorch 1.12.1.

4.2. Comparisons with the State-of-the-Art FER Methods

We compare the performance of SSPA-JT with various state-of-the-art methods on two target datasets, RAF-DB and CAER-S, using ResNet18 [49] and ARM [51] as backbones. For the experiment on RAF-DB, SSPA-JT employed 10% of AffectNet as the auxiliary dataset, while for the experiment on CAER-S, 20% of AffectNet was used. The reason for using specific proportions of AffectNet as the auxiliary dataset for SSPA-JT is to avoid the performance degradation caused by introducing too much auxiliary data, which will be discussed in detail in Section 4.6.

As shown in Table 1, SSPA-JT obtained the highest accuracy of 92.44% on RAF-DB. Compared with other joint training methods, such as IPA2LT [15], SCN [19], and DCJT [18], which also incorporate extra data during training, SSPA-JT outperformed all in recognition accuracy, which highlights its superior efficiency in using auxiliary data. Notably, while other approaches like APViT [54] and LA-Net [37] have achieved high accuracies of 91.98% and 91.56%, respectively, by employing different strategies to enhance FER performance, SSPA-JT still surpassed them. This suggests that our joint training framework, combined with the careful management of auxiliary data, provides a competitive advantage in FER tasks. Among the compared methods, ARM [51] achieved an accuracy of 90.42% without using any extra data because of its strong feature extraction capability. Including samples from auxiliary datasets into the training process with the methods like DCJT and SSPA-JT further improves the performance of ARM.

Table 1. Comparison with state-of-the-art methods on RAF-DB. SSPA-JT uses ARM as the backbone, RAF-DB as the target dataset, and 10% of AffectNet as the auxiliary dataset. The best and second best are bolded and underlined, respectively.

Backbone	Acc. (%)	
IPA2LT [15]	86.77	With extra data
SCN [19]	88.14	With extra data
RUL [34]	88.98	
ARM [51]	90.42	
EASE [35]	89.56	
EAC [36]	89.99	
APViT [54]	91.98	
LA-Net [37]	91.56	
DCJT [18]	<u>92.24</u>	With extra data
SSPA-JT	<u>92.44</u>	With extra data

Table 2 compares our method with multiple state-of-the-art methods on the CAER-S test set. As the table indicates, SSPA-JT achieved the highest accuracy of 98.22%, significantly outperforming all other methods. This remarkable result further highlights the effectiveness of our joint training framework in improving FER performance. Overall, the comparison demonstrates that SSPA-JT is highly effective and achieves state-of-the-art results, making it a promising approach for improving FER performance through the integration of auxiliary data.

Table 2. Comparison with state-of-the-art method on CAER-S. SSPA-JT uses ARM as the backbone, CAER-S as the target dataset, and 20% of AffectNet as the auxiliary dataset. The best and second best are bolded and underlined, respectively.

Backbone	Acc. (%)	
ResNet18 [49]	84.67	
ResNet50 [49]	84.81	
CAER-NET-S [12]	73.51	
Res2Net [55]	85.35	
MANET [56]	88.42	
ARM [51]	91.54	
EASE [35]	90.95	
EAC [18,36]	91.33	
DCJT [18]	<u>94.57</u>	With extra data
SSPA-JT	98.22	With extra data

Overall, the comparison demonstrates that SSPA-JT is highly effective and achieves state-of-the-art results, making it a promising approach for improving FER performance through the integration of auxiliary data.

4.3. Comparison with Other Joint Training Methods

To further demonstrate the effectiveness of SSPA-JT, we conducted additional experiments to specifically compare its performance with other FER joint training methods using RAF-DB and CAER-S as the target datasets. We employed ResNet18 and ARM as backbones and compared SSPA-JT with four other joint training strategies: (1) training solely on the target dataset without extra datasets; (2) training on the dataset that combines the target and auxiliary datasets; (3) the training strategy of SCN, which employs uncertainty-based weighting for sample learning; and (4) the training strategy of DCJT, which aligns different datasets using both discrete and continuous labels. Similar to the setting in Section 4.2, SSPA-JT used 10% of AffectNet as the auxiliary dataset for the experiment on RAF-DB and 20% for CAER-S. In contrast, other methods employed the entire AffectNet dataset as the auxiliary dataset by default. The experimental results are presented in Tables 3 and 4.

Table 3. Comparison of different joint training methods on the test set of RAF-DB based on the ARM and ResNet18 backbones. * denotes that all of AffectNet is used as \mathcal{D}_A , [†] denotes only 10% of AffectNet is used as \mathcal{D}_A . The best and second best are bolded and underlined, respectively.

Backbone	Joint-Learning Method	Acc. (%)
ResNet18	Without auxiliary dataset [18]	86.25
	Combination straightly * [18]	84.62
	SCN * [19]	88.14
	DCJT * [18]	88.48
	SSPA-JT [†]	<u>88.23</u>
ARM	Without auxiliary dataset [18]	90.42
	Combination straightly * [18]	89.27
	SCN * [18,19]	91.06
	DCJT * [18]	<u>92.24</u>
	SSPA-JT [†]	92.44

When evaluated on the test set of RAF-DB with ResNet18 as the backbone, the strategy trained solely on the target dataset achieved an accuracy of 86.25%. However, combining the two FER datasets directly for joint training led to a 0.63% decrease in accuracy compared to using only the target dataset. The SCN method improved these results, reaching an accuracy of 88.14%, which was a significant enhancement over the previous two methods without any special design. DCJT further increased the accuracy to 88.48%, achieving the best result with ResNet18. SSPA-JT obtained the accuracy of 88.23%, which was slightly

lower than the performance of DCJT. For the experiments using ARM as the backbone, a similar trend was observed. The model trained solely on the target dataset obtained an accuracy of 90.42%. Directly combining the datasets caused a 1.15% drop in accuracy. SCN improved this to 91.06%, and DCJT obtained a higher accuracy of 92.24%. Notably, SSPA-JT achieved comparable accuracy (92.44%) with DCJT.

Similar results were observed from the experiment on the CAER-S dataset. These results demonstrate that SSPA-JT effectively enhances the performance of FER joint training particularly when using backbones with stronger feature extraction capabilities.

Table 4. Comparison of different joint training methods on the test set of CAER-S based on the ARM, ResNet18 backbones. * denotes that all of AffectNet is used as \mathcal{D}_A , [†] denotes only 20% of AffectNet is used as \mathcal{D}_A . The best and second best are bolded and underlined, respectively.

Backbone	Joint-Learning Method	Acc. (%)
ResNet18	Without auxiliary dataset [18]	84.67
	Combination straightly * [18]	81.45
	SCN * [18,19]	84.31
	DCJT * [18]	86.39
	SSPA-JT [†]	<u>84.69</u>
ARM	Without auxiliary dataset [18]	91.54
	Combination straightly * [18]	84.36
	SCN * [18,19]	90.39
	DCJT * [18]	<u>94.57</u>
	SSPA-JT [†]	98.22

4.4. Ablation Study

To further explore the contributions of each module in the SSPA-JT framework, we conducted ablation experiments to assess the impact of individual components on overall performance. We focused on four key components of SSPA-JT: (1) Sample Selection Module (SSM), (2) fixing the target dataset to clean set (\mathcal{D}_T fixed), (3) Mixture of Experts module (MoE), and (4) Paired Augmentation Module (PAM). The experiments were carried out using ResNet18 and ARM as backbones, RAF-DB as the target dataset and 10% of AffectNet as the auxiliary dataset. In addition to accuracy, we evaluated the performance in terms of Macro-Precision, Macro-Recall, and Macro-F1 Score to provide a more comprehensive assessment of the model. The experimental results of the ablation study are shown in Table 5.

As shown in Table 5, joint training based on a direct combination of the target and auxiliary datasets, i.e., none of the evaluated modules were employed, leads to a limited performance, with accuracy being lower than that achieved by training on the target dataset alone. Introducing the SSM module resulted in a slight improvement in accuracy, indicating that the sample selection module enhances the quality of the dataset and improves the performance of the model. However, in this case, the target dataset was not fixed as clean set, the standard for samples selection during joint training was ambiguous, making the selection process disturbed by the auxiliary dataset, which affects the effectiveness of the selection.

When both SSM and fixed \mathcal{D}_T were employed, the accuracy of the model on both backbones slightly exceeded the accuracy obtained by training solely on the target dataset. This suggests that selecting samples from the auxiliary dataset in accordance with the annotation standard of the target dataset improves the joint training and model performance. The combination of SSM and fixed \mathcal{D}_T is particularly effective because it establishes a clear standard for sample selection, making the process more targeted. As shown in Figure 6, we provided visual examples of samples with consistent annotations and those with inconsistent annotations, offering an intuitive understanding of the impact of SSM.

Table 5. Ablation study on RAF-DB as the target dataset \mathcal{D}_T and 10% of AffectNet as the auxiliary dataset \mathcal{D}_A using ResNet18 and ARM backbones. \mathcal{D}_T fixed denotes always fixing \mathcal{D}_T to the clean set \mathcal{D}_C . Macro-P denotes Macro-Precision, Macro-R denotes Macro-Recall, and Macro-F1 denotes Macro-F1 Score. All results are presented as percentages. The best and second best are bolded and underlined, respectively.

Backbone	Modules				Metrics			
	SSM	\mathcal{D}_T Fixed	MoE	PAM	Acc.	Macro-P	Macro-R	Macro-F1
ResNet18					85.06	78.36	75.93	76.92
	✓				86.53	82.53	77.07	78.92
	✓	✓			86.76	81.10	78.31	79.37
	✓		✓		86.51	83.99	75.95	78.15
	✓			✓	85.27	79.84	74.64	75.67
	✓	✓	✓		<u>87.35</u>	<u>83.02</u>	78.67	80.28
	✓	✓		✓	87.28	81.52	<u>79.99</u>	<u>80.51</u>
	✓		✓	✓	83.71	78.02	68.19	68.08
ARM					89.84	84.24	83.65	83.91
	✓				90.29	85.32	82.94	83.85
	✓	✓			90.75	85.10	83.87	84.00
	✓		✓		90.03	84.89	81.53	82.80
	✓		✓	✓	87.53	84.11	75.44	77.50
	✓	✓	✓		91.33	85.94	85.40	85.61
	✓	✓		✓	<u>91.63</u>	<u>88.41</u>	<u>86.10</u>	<u>87.14</u>
	✓		✓	✓	88.41	83.94	76.80	79.05
	✓	✓	✓	92.44	89.18	86.46	87.66	

Incorporating MoE or PAM alongside SSM and fixed \mathcal{D}_T led to further accuracy improvements. The MoE module increased the model capacity and provided SSM with diverse predictions, which allowed the model to better capture the characteristics of each sample and improved the selection of samples with consistent annotation. Table 5 also shows that PAM alleviated the class bias caused by the imbalance in the FER datasets and enhanced the overall model performance. In the case \mathcal{D}_T was fixed, training incorporating PAM consistently led to significantly higher Macro-Precision, Macro-Recall, and Macro-F1 Score values compared to other designs.

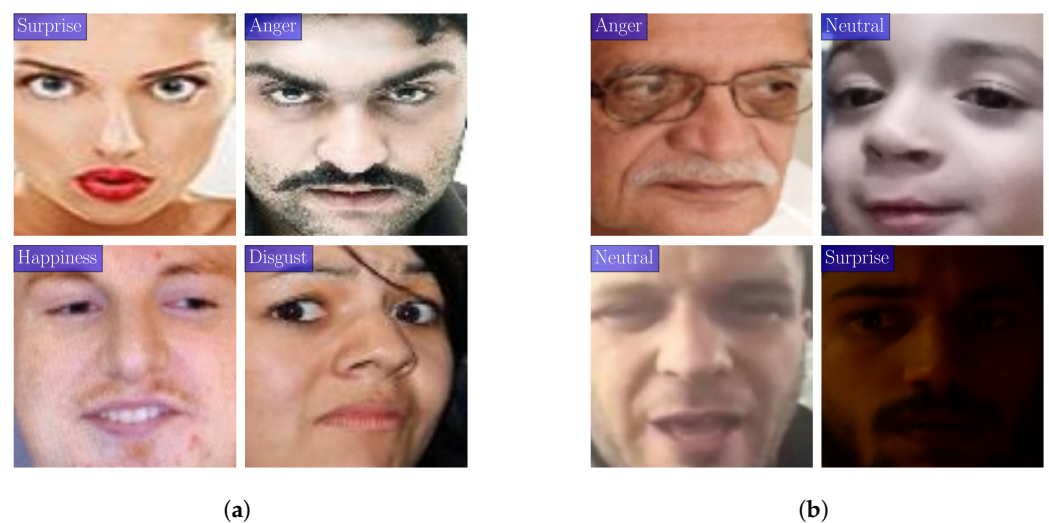


Figure 6. Visualization of sample selection via SSM at the final epoch. (a) Samples with consistent annotations; (b) samples with inconsistent annotations.

When all modules were employed, forming the complete SSPA-JT, the model achieved the best performance on both backbones. The ablation experiments demonstrate that

each module in SSPA-JT plays a crucial role, working synergistically to produce superior performance. These results also indicate that addressing both label inconsistency and class imbalance is essential to significantly improve the effectiveness of joint training on FER datasets.

4.5. Hyperparameter Experiments

To analyze the sensitivity of SSPA-JT against the variation of the hyperparameters, we conducted experiments where we set different values to the number of experts k in the MoE and the parameters α and β of the beta distribution used for Mixup in the PAM. The experiments were carried out using ARM as backbones, RAF-DB as the target dataset and 10% of AffectNet as the auxiliary dataset.

Table 6 shows the impact of the number of experts in MoE. When the value of k was set to 3, SSPA-JT obtained the best performance in all metrics. However, the performance decreased when k was either too large or too small. Our opinion is that a small k limits the capacity of model to leverage diverse expert opinions, which leads to underfitting. On the other hand, a large k may introduce redundancy and increase the complexity of the model, which results in overfitting or difficulty in effectively combining the outputs of multiple experts. Therefore, a moderate value of k strikes a balance between model complexity and representational capacity, provides richer information for sample selection, and leads to improved performance.

Table 6. The results of SSPA-JT using different numbers of experts in MoE. The best and second best are bolded and underlined, respectively.

k	Acc.	Macro-P	Macro-R	Macro-F1
1	91.34	87.08	<u>84.77</u>	85.79
2	<u>91.63</u>	87.69	84.39	<u>85.86</u>
3	92.44	89.18	86.46	87.66
5	91.47	<u>87.70</u>	84.12	85.69
10	90.46	85.69	80.93	82.92

Table 7 indicates the impact of different beta distribution parameters α and β in the PAM. It shows that SSTA-JT obtained the highest Accuracy, Macro-Recall, and Macro-F1 Score, and it obtained the second highest Macro-Accuracy when α and β were set to 5 and 1, respectively. The performance declined when α or β values deviated significantly from this configuration. For example, the performance metrics decreased when α was set to 10 and β to 1, since Mixup does not provide sufficient augmentation diversity by producing samples similar to the original samples. On the other hand, when α was set to 1 and β to 10, the generated samples may be too noisy or dissimilar, leading to degraded performance. Therefore, it is crucial to find an appropriate balance in the beta distribution parameters to achieve effective augmentation and improved model performance.

Table 7. The results of SSPA-JT using different beta distribution in PAM. The best and second best are bolded and underlined, respectively.

β	α	Acc.	Macro-P	Macro-R	Macro-F1
1	1	91.79	88.21	83.97	85.81
	3	91.53	87.69	83.96	85.60
	5	92.44	<u>89.18</u>	86.46	87.66
	10	91.24	85.99	84.70	85.27
3	1	91.14	86.64	84.48	85.47
	3	91.79	89.09	84.51	86.55
	5	91.43	87.65	84.22	85.78
	10	91.82	87.84	85.00	86.28

Table 7. Cont.

β	α	Acc.	Macro-P	Macro-R	Macro-F1
5	1	90.36	86.08	83.85	84.72
	3	91.50	88.14	84.40	86.07
	5	91.34	87.20	83.48	85.17
	10	91.82	88.43	84.57	86.27
10	1	91.11	86.08	<u>85.08</u>	85.36
	3	91.05	87.01	84.38	85.50
	5	91.34	86.59	83.93	85.14
	10	<u>91.86</u>	89.25	84.75	<u>86.73</u>

4.6. Discussion on the Impact of Auxiliary Dataset Scale

In our experiments, we observed that the scale of the auxiliary dataset significantly impacted the performance of SSPA-JT. Contrary to common assumptions, a larger auxiliary dataset does not always lead to better performance. To further investigate this pattern in detail, we conducted experiments using ARM as the backbone, with RAF-DB as the target dataset and varying the proportions of AffectNet as the auxiliary dataset.

Table 8 shows the accuracy of SSPA-JT using different proportions of the AffectNet training set as the auxiliary dataset while keeping RAF-DB as the target dataset. The data reveal that SSPA-JT achieved the best performance when the scale of the auxiliary dataset was comparable to or slightly larger than that of the target dataset. Specifically, with ResNet18 as the backbone, the best accuracy (88.23%) was obtained when 10% of AffectNet was used. Once the proportion increased beyond this point, the performance of SSPA-JT declined. For example, the accuracy of SSPA-JT dropped to 83.50% when 100% of AffectNet was used. This suggests that a moderate amount of auxiliary data provides beneficial information. Too much auxiliary data may introduce additional noise or domain discrepancies that hinders generalization.

Table 8. Results of SSPA-JT on RAF-DB as the target dataset and different proportions of AffectNet as the auxiliary dataset. Percentage denotes the proportion of AffectNet training set used. The best is bolded.

Backbone	Percentage (%)	Acc.
ResNet18	0	86.25
	3	87.18
	5	87.41
	10	88.23
	20	86.36
	50	85.88
	100	83.50
ARM	0	90.42
	3	91.47
	5	91.99
	10	92.44
	20	91.57
	100	89.09

A similar trend is observed when using ARM as the backbone with the highest accuracy of 92.44% achieved when 10% of AffectNet was used. Increasing the amount of auxiliary data beyond this scale again led to a decline in performance.

To further illustrate the effectiveness of SSPA-JT, we also conducted experiments where the target and auxiliary datasets were directly merged for joint training without the SSPA-JT framework. The results of these experiments are shown in Table 9. For both backbones, the best performance was achieved when the target dataset was used alone without any auxiliary data. In the case of ResNet18, the performance consistently deteriorated as the

size of the auxiliary dataset increased. Although the trend is less pronounced for ARM, the direct merging of datasets still led to a decline in performance. This might result from the powerful feature extraction capability of ARM, which makes ARM more robust to label noise introduced by annotation inconsistencies among datasets. However, even with ARM, a large auxiliary dataset without proper handling also led to degraded performance.

Experimental results of Tables 8 and 9 indicate choosing an appropriate proportion of the auxiliary dataset and employing a robust joint training framework like SSPA-JT are crucial for achieving improved FER performance.

Table 9. Results of directly merging RAF-DB with different proportions of AffectNet. Percentage denotes the proportion of AffectNet training set used. The best is bolded.

Backbone	Percentage (%)	Acc.
ResNet18	0	86.25
	3	86.21
	5	86.04
	10	85.06
	20	84.87
	50	84.97
	100	84.62
ARM	0	90.42
	3	89.52
	5	89.71
	10	89.84
	20	89.48
	50	89.42
	100	89.27

5. Conclusions

In this study, we propose the SSPA-JT framework to address the challenges of annotation inconsistency and class imbalance in multi-dataset joint training for FER. By modeling annotation inconsistency as a problem of label noise and incorporating sample selection alongside a dynamic matching algorithm for tail class augmentation, our method effectively mitigates the negative impact of these issues on model performance. Experimental results confirm that SSPA-JT not only improves FER accuracy but also enhances the robustness of models against background noise and class bias. The proposed framework offers a viable solution to improve the effectiveness of joint training across multiple datasets with diverse annotation standards and imbalanced class distributions.

Future work could focus on adapting the SSPA-JT framework for additional domains, such as emotion recognition in video or integrating multi-modal data. Exploring its potential in combination with advanced techniques like self-supervised learning and adaptive data augmentation could further enhance its performance and adaptability. These directions hold promise for advancing the field and developing more robust and versatile FER systems.

Author Contributions: T.C.: conceptualization, methodology, software, validation, formal analysis, investigation, resources, data curation, writing—original draft preparation, visualization; D.Z.: curation, writing—original draft preparation, writing—review and editing, supervision, project administration, funding acquisition; D.-J.L.: investigation, writing—review and editing, supervision. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by National Natural Science Foundation of China (62173353), Science and Technology Program of Guangzhou, China (2023B03J1327), and Guangdong Science and Technology Program (2023A1111120012).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The Real-world Affective Faces Database (RAF-DB) is accessible at GitHub, <http://whdeng.cn/RAF/model1.html#dataset> (accessed on 9 September 2024). The Real-world Affective Faces Database (RAF-DB) is accessible at GitHub, <https://caer-dataset.github.io/> (accessed on 9 September 2024). The Context-Aware Emotion Recognition dataset (CAER-S) is accessible at <http://mohammadmahoor.com/affectnet/> (accessed on 9 September 2024). The data generated during the current study are available from the corresponding author on reasonable request.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Chowdary, M.K.; Nguyen, T.N.; Hemanth, D.J. Deep Learning-Based Facial Emotion Recognition for Human–Computer Interaction Applications. *Neural Comput. Appl.* **2023**, *35*, 23311–23328. [CrossRef]
2. Wang, Y.; Song, W.; Tao, W.; Liotta, A.; Yang, D.; Li, X.; Gao, S.; Sun, Y.; Ge, W.; Zhang, W.; et al. A Systematic Review on Affective Computing: Emotion Models, Databases, and Recent Advances. *Inf. Fusion* **2022**, *83–84*, 19–52. [CrossRef]
3. Muhammad, G.; Alsulaiman, M.; Amin, S.U.; Ghoneim, A.; Alhamid, M.F. A Facial-Expression Monitoring System for Improved Healthcare in Smart Cities. *IEEE Access* **2017**, *5*, 10871–10881. [CrossRef]
4. Munsif, M.; Ullah, M.; Ahmad, B.; Sajjad, M.; Cheikh, F.A. Monitoring Neurological Disorder Patients via Deep Learning Based Facial Expressions Analysis. In Proceedings of the Artificial Intelligence Applications and Innovations, AIAI 2022 IFIP WG 12.5 International Workshops, Crete, Greece, 17–20 June 2022; Maglogiannis, I., Iliadis, L., Macintyre, J., Cortez, P., Eds.; 2022; pp. 412–423.
5. Munsif, M.; Sajjad, M.; Ullah, M.; Tarekegn, A.N.; Cheikh, F.A.; Tsakanikas, P.; Muhammad, K. Optimized Efficient Attention-Based Network for Facial Expressions Analysis in Neurological Health Care. *Comput. Biol. Med.* **2024**, *179*, 108822. [CrossRef] [PubMed]
6. Lucey, P.; Cohn, J.F.; Kanade, T.; Saragih, J.; Ambadar, Z.; Matthews, I. The Extended Cohn-Kanade Dataset (CK+): A Complete Dataset for Action Unit and Emotion-Specified Expression. In Proceedings of the 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition-Workshops, San Francisco, CA, USA, 13–18 June 2010; pp. 94–101.
7. Dhall, A.; Goecke, R.; Lucey, S.; Gedeon, T. Static Facial Expression Analysis in Tough Conditions: Data, Evaluation Protocol and Benchmark. In Proceedings of the 2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops), Barcelona, Spain, 6–13 November 2011; pp. 2106–2112.
8. Lyons, M.; Akamatsu, S.; Kamachi, M.; Gyoba, J. Coding Facial Expressions with Gabor Wavelets. In Proceedings of the Third IEEE International Conference on Automatic Face and Gesture Recognition, Nara, Japan, 14–16 April 1998; pp. 200–205.
9. Li, S.; Deng, W.; Du, J. Reliable Crowdsourcing and Deep Locality-Preserving Learning for Expression Recognition in the Wild. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–16 July 2017; pp. 2584–2593.
10. Li, S.; Deng, W. Reliable Crowdsourcing and Deep Locality-Preserving Learning for Unconstrained Facial Expression Recognition. *IEEE Trans. Image Process.* **2019**, *28*, 356–370. [CrossRef] [PubMed]
11. Mollahosseini, A.; Hasani, B.; Mahoor, M.H. AffectNet: A Database for Facial Expression, Valence, and Arousal Computing in the Wild. *IEEE Trans. Affect. Comput.* **2019**, *10*, 18–31. [CrossRef]
12. Lee, J.; Kim, S.; Kim, S.; Park, J.; Sohn, K. Context-Aware Emotion Recognition Networks. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019; pp. 10142–10151.
13. Zhang, Z.; Luo, P.; Loy, C.C.; Tang, X. From Facial Expression Recognition to Interpersonal Relation Prediction. *Int. J. Comput. Vis.* **2018**, *126*, 550–569. [CrossRef]
14. Benitez-Quiroz, C.F.; Srinivasan, R.; Martinez, A.M. EmotioNet: An Accurate, Real-Time Algorithm for the Automatic Annotation of a Million Facial Expressions in the Wild. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 5562–5570.
15. Sun, C.; Shrivastava, A.; Singh, S.; Gupta, A. Revisiting Unreasonable Effectiveness of Data in Deep Learning Era. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 843–852.
16. Zeng, J.; Shan, S.; Chen, X. Facial Expression Recognition with Inconsistently Annotated Datasets. In Proceedings of the Computer Vision–ECCV Munich, Germany, 8–14 September 2018; Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y., Eds.; pp. 227–243.
17. Van Horn, G.; Perona, P. The Devil Is in the Tails: Fine-grained Classification in the Wild. *arXiv* **2017**, arXiv:1709.01450.
18. Yu, C.; Zhang, D.; Zou, W.; Li, M. Joint Training on Multiple Datasets With Inconsistent Labeling Criteria for Facial Expression Recognition. *IEEE Trans. Affect. Comput.* **2024**, *15*, 1812–1825. [CrossRef]
19. Wang, K.; Peng, X.; Yang, J.; Lu, S.; Qiao, Y. Suppressing Uncertainties for Large-Scale Facial Expression Recognition. *arXiv* **2020**, arXiv:2002.10392.
20. Li, Y.; Han, H.; Shan, S.; Chen, X. DISC: Learning from Noisy Labels via Dynamic Instance-Specific Selection and Correction. In Proceedings of the 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, BC, Canada, 17–24 June 2023; pp. 24070–24079.
21. Shan, C.; Gong, S.; McOwan, P.W. Facial Expression Recognition Based on Local Binary Patterns: A Comprehensive Study. *Image Vis. Comput.* **2009**, *27*, 803–816. [CrossRef]

22. Dalal, N.; Triggs, B. Histograms of Oriented Gradients for Human Detection. In Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), San Diego, CA, USA, 20–26 June 2005; Volume 1, pp. 886–893.
23. Lowe, D.G. Distinctive Image Features from Scale-Invariant Keypoints. *Int. J. Comput. Vis.* **2004**, *60*, 91–110. [[CrossRef](#)]
24. Zhang, C.; Bengio, S.; Hardt, M.; Recht, B.; Vinyals, O. Understanding Deep Learning (Still) Requires Rethinking Generalization. *Commun. ACM* **2021**, *64*, 107–115. [[CrossRef](#)]
25. Karim, N.; Rizve, M.N.; Rahnvard, N.; Mian, A.; Shah, M. UniCon: Combating Label Noise through Uniform Selection and Contrastive Learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 9676–9686.
26. Jiang, L.; Zhou, Z.; Leung, T.; Li, L.J.; Li, F.-F. MentorNet: Learning Data-Driven Curriculum for Very Deep Neural Networks on Corrupted Labels. In Proceedings of the 35th International Conference on Machine Learning, PMLR, Stockholm, Sweden, 10–15 July 2018; pp. 2304–2313.
27. Cheng, H.; Zhu, Z.; Li, X.; Gong, Y.; Sun, X.; Liu, Y. Learning with Instance-Dependent Label Noise: A Sample Sieve Approach. In Proceedings of the International Conference on Learning Representations, Vienna, Austria, 4 May 2021.
28. Han, B.; Yao, Q.; Yu, X.; Niu, G.; Xu, M.; Hu, W.; Tsang, I.W.; Sugiyama, M. Co-Teaching: Robust Training of Deep Neural Networks with Extremely Noisy Labels. In Proceedings of the 32nd International Conference on Neural Information Processing Systems, NIPS'18, Montreal, QC, Canada, 3–8 December 2018; Curran Associates Inc.: Red Hook, NY, USA, 2018; pp. 8536–8546.
29. Tanaka, D.; Ikami, D.; Yamasaki, T.; Aizawa, K. Joint Optimization Framework for Learning with Noisy Labels. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 5552–5560.
30. Li, J.; Xiong, C.; Hoi, S.C. Learning from Noisy Data with Robust Representation Learning. In Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, BC, Canada, 11–17 October 2021; pp. 9465–9474.
31. Zhang, Z.; Sabuncu, M.R. Generalized Cross Entropy Loss for Training Deep Neural Networks with Noisy Labels. In Proceedings of the 32nd International Conference on Neural Information Processing Systems, NIPS'18, Montreal, QC, Canada, 3–8 December 2018; Curran Associates Inc.: Red Hook, NY, USA, 2018; pp. 8792–8802.
32. Engleson, E.; Azizpour, H. Generalized Jensen-Shannon Divergence Loss for Learning with Noisy Labels. In Proceedings of the 35th International Conference on Neural Information Processing Systems, NIPS '21, Online, 6–14 December 2021; Curran Associates Inc.: Red Hook, NY, USA, 2024; pp. 30284–30297.
33. Arazo, E.; Ortego, D.; Albert, P.; O'Connor, N.; McGuinness, K. Unsupervised Label Noise Modeling and Loss Correction. In Proceedings of the 36th International Conference on Machine Learning, PMLR, Long Beach, CA, USA, 9–15 June 2019; pp. 312–321.
34. Zhang, Y.; Wang, C.; Deng, W. Relative Uncertainty Learning for Facial Expression Recognition. In Proceedings of the Advances in Neural Information Processing Systems, Online, 6–14 December 2021.
35. Wang, L.; Jia, G.; Jiang, N.; Wu, H.; Yang, J. EASE: Robust Facial Expression Recognition via Emotion Ambiguity-Sensitive Cooperative Networks. In Proceedings of the 30th ACM International Conference on Multimedia, MM '22, Lisboa, Portugal, 10–14 October 2022; ACM: New York, NY, USA, 2022; pp. 218–227.
36. Zhang, Y.; Wang, C.; Ling, X.; Deng, W. Learn from All: Erasing Attention Consistency for Noisy Label Facial Expression Recognition. In Proceedings of the Computer Vision—ECCV, Tel Aviv, Israel, 23–27 October 2022; Avidan, S., Brostow, G., Cissé, M., Farinella, G.M., Hassner, T., Eds.; 2022; pp. 418–434.
37. Wu, Z.; Cui, J. LA-Net: Landmark-Aware Learning for Reliable Facial Expression Recognition under Label Noise. In Proceedings of the 2023 IEEE/CVF International Conference on Computer Vision (ICCV), Paris, France, 1–6 October 2023; pp. 20641–20650.
38. Kang, B.; Xie, S.; Rohrbach, M.; Yan, Z.; Gordo, A.; Feng, J.; Kalantidis, Y. Decoupling Representation and Classifier for Long-Tailed Recognition. In Proceedings of the Eighth International Conference on Learning Representations, Addis Ababa, Ethiopia, 26–30 April 2020.
39. Hu, X.; Jiang, Y.; Tang, K.; Chen, J.; Miao, C.; Zhang, H. Learning to Segment the Tail. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 14042–14051.
40. Lin, T.Y.; Goyal, P.; Girshick, R.; He, K.; Dollar, P. Focal Loss for Dense Object Detection. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2980–2988.
41. Cui, Y.; Jia, M.; Lin, T.Y.; Song, Y.; Belongie, S. Class-Balanced Loss Based on Effective Number of Samples. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 9260–9269.
42. Du, F.; Yang, P.; Jia, Q.; Nan, F.; Chen, X.; Yang, Y. Global and Local Mixture Consistency Cumulative Learning for Long-tailed Visual Recognitions. In Proceedings of the 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, BC, Canada, 17–24 June 2023; pp. 15814–15823.
43. Li, S.; Gong, K.; Liu, C.H.; Wang, Y.; Qiao, F.; Cheng, X. MetaSAug: Meta Semantic Augmentation for Long-Tailed Visual Recognition. In Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021; pp. 5208–5217.

44. Park, S.; Hong, Y.; Heo, B.; Yun, S.; Choi, J.Y. The Majority Can Help the Minority: Context-rich Minority Oversampling for Long-tailed Classification. In Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022; pp. 6877–6886.
45. Gao, H.; An, S.; Li, J.; Liu, C. Deep Balanced Learning for Long-tailed Facial Expressions Recognition. In Proceedings of the 2021 IEEE International Conference on Robotics and Automation (ICRA), Xi'an, China, 30 May–5 June 2021; pp. 11147–11153.
46. Zhang, Y.; Li, Y.; Qin, L.; Liu, X.; Deng, W. Leave No Stone Unturned: Mine Extra Knowledge for Imbalanced Facial Expression Recognition. *arXiv* **2023**, arXiv:2310.19636.
47. Zhang, H.; Cisse, M.; Dauphin, Y.N.; Lopez-Paz, D. Mixup: Beyond Empirical Risk Minimization. *arXiv* **2018**, arXiv:1710.09412.
48. Kraft, D. Algorithm 733: TOMP–Fortran Modules for Optimal Control Calculations. *ACM Trans. Math. Softw.* **1994**, *20*, 262–281. [[CrossRef](#)]
49. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 26 June–1 July 2016; pp. 770–778.
50. Guo, Y.; Zhang, L.; Hu, Y.; He, X.; Gao, J. MS-Celeb-1M: A Dataset and Benchmark for Large-Scale Face Recognition. In Proceedings of the Computer Vision–ECCV, Amsterdam, The Netherlands, 11–14 October 2016; Leibe, B., Matas, J., Sebe, N., Welling, M., Eds.; 2016; pp. 87–102.
51. Shi, J.; Zhu, S.; Liang, Z. Learning to Amend Facial Expression Representation via De-albino and Affinity. *arXiv* **2021**, arXiv:2103.10189.
52. Deng, J.; Dong, W.; Socher, R.; Li, L.J.; Li, K.; Li, F.-F. ImageNet: A Large-Scale Hierarchical Image Database. In Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; pp. 248–255.
53. Loshchilov, I.; Hutter, F. SGDR: Stochastic Gradient Descent with Warm Restarts. *arXiv* **2017**, arXiv:1608.03983.
54. Xue, F.; Wang, Q.; Tan, Z.; Ma, Z.; Guo, G. Vision Transformer With Attentive Pooling for Robust Facial Expression Recognition. *IEEE Trans. Affect. Comput.* **2022**, *14*, 3244–3256. [[CrossRef](#)]
55. Gao, S.H.; Cheng, M.M.; Zhao, K.; Zhang, X.Y.; Yang, M.H.; Torr, P. Res2Net: A New Multi-Scale Backbone Architecture. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *43*, 652–662. [[CrossRef](#)]
56. Zhao, Z.; Liu, Q.; Wang, S. Learning Deep Global Multi-Scale and Local Attention Features for Facial Expression Recognition in the Wild. *IEEE Trans. Image Process.* **2021**, *30*, 6544–6556. [[CrossRef](#)] [[PubMed](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.