ABSTRACT The sample imbalance of expression datasets always leads to poor recognition results for minority classes. To solve this problem, we propose a facial expression recognition network, called Residual Attentive Sharing Network (RASN). There is a fact that different expressions have affinity features, which makes it possible for the minority classes to benefit from the majority classes in the expression feature extraction process, from which we propose a sharing affinity features module to compensate for the inadequate feature learning of minority classes by sharing affinity features. In addition, an affinity features attention module is added to highlight expression-related affinity features and suppress expression-unrelated ones for enhancing the role of sharing affinity features. Experiments on the CK+ , RAF-DB, and FER2013 datasets validate the robustness of our method to sample imbalance. The validation accuracies of our method are 96.97% on CK+, 71.44% on FER2013, and 90.91% on RAF-DB, respectively, which exceed current state-of-the-art methods.

INDEX TERMS Facial expression recognition, affinity features, attention mechanism, residual attentive sharing network.

I. INTRODUCTION
Facial Expression Recognition (FER) is a vital research topic in a wide range of research fields ranging from artificial intelligence to Psychology. With the improvement of society automation, the applications of FER have been gradually increasing in the fields of security, medical, criminal investigation, and education. Conventional methods use hand-crafted features [1], [2], [3], [4] to achieve expression classification. However, the hand-crafted features are only human-designed features, which have weak representational power and lack the ability to accurately express semantic information. This leads to the poor performance of conventional methods on FER task. In recent years, with the booming development of deep learning, various deep learning FER methods have been proposed. As the number of network layers increases, deep neural networks [5], [6], [7] have more representation ability and can express semantic information more accurately. The deep learning methods have the advantage of higher recognition accuracies over conventional methods.

However, it is difficult for the deep learning methods to achieve the same enhancement effect on datasets, such as RAF-DB [8] and FER2013 [9]. This is because humans express expressions at different frequencies in real scenes, resulting in different difficulties in collecting different expressions. As shown in Figure 1, the distribution of the number of expressions of each category on RAF-DB and FER2013 dataset is extremely unbalanced, what is called sample imbalance. The phenomenon will lead to insufficient feature learning for the minority classes and reduce the recognition accuracy. Recent studies have proposed different methods to address the sample imbalance issue in FER. Data augmentation is the most common method.
Xie et al. [10] propose a TDGAN, which generates minority class samples for data augmentation by using Generative Adversarial Networks to reduce the impact of sample imbalance on the classification effect. Contrary to it, Shi et al. [11] propose the concept of affinity features for facial expressions based on the fact that FER is a classification task for faces, and reduce the effect of sample imbalance on the model by sharing affinity features. However, Shi et al. [11] still suffer from sample imbalance, which we believe is mainly due to the fact that the sharing affinity features module proposed by Shi et al. don’t take the channel information into account when finding affinity features and simply sharing affinity features in the last layer of feature layers.

To solve the problem, we propose a Residual Attention Sharing Network (RASN) by combining improved sharing affinity features and new attention mechanism. Specifically, we add four sharing affinity features modules (Sharing-module) and affinity features attention modules (Attention-module) in the four feature extraction layers of the residual network [7] to achieve multiple times sharing. Compared with the original sharing affinity features module, our Sharing-module obtains affinity features that preserve channel information by averaging the original features of batch samples. The extracted affinity features are added to the original features element-wisely to change the learning route for both minority and majority classes. This enables the model to benefit from the majority class expressions when learning features for the minority class expressions, which offers the possibility to compensate for the inadequate features learning of minority class expressions due to the insufficient number. On the base of which the affinity features retain channel information, Attention-module learns a weight for each channel of affinity features to obtain the importance of affinity features. It expects to assign higher weights to expression-related affinity features and lower weights to expression-unrelated ones. The learned weights are multiplied by affinity features to highlight expression-related affinity features and suppress expression-unrelated ones, which can further enhance the role of sharing affinity features.

The main contributions of this paper are summarized as follows:

1. A novel RASN is proposed to solve the sample imbalance in expression recognition by sharing affinity features in the four feature extraction layers of the ResNet18.

2. An affinity features attention module is proposed to learn the importance weights of affinity features. So as to further enhance the role of sharing affinity features.

3. The robustness of our RASN against sample imbalance is validated on the CK+, FER2013, and RAF-DB datasets. And the validation accuracies of our method are 96.97% on CK+, 71.44% on FER2013, and 90.91% on RAF-DB, respectively.

II. RELATED WORK

A. FACIAL EXPRESSION RECOGNITION

In the 1970s, American psychologist Ekmen defined seven categories of facial expressions: happy, angry, surprise, fear, disgust, sad, and neutral, respectively, which determined the categories of recognizing objects. In order to identify the category of facial expression based on a face image, usually comprise three processes, namely face image pretreatment, facial expression feature extraction, and facial expression recognition. According to the types of extracted features, facial expression recognition methods can be grouped into hand-crafted features and learning-based features. For the hand-crafted features, they mainly include Local Binary Pattern features [1], Scale Invariant Transform features [2], Histogram Of Gradient features Local Binary Pattern features [3], Gobar features as well as Histogram Of Gradient features [4]. For the learning-based features, Meng et al. [12] realize a frame attention network with ResNet18 as the backbone network to make the model pay closer attention to important frames and aim at improving the ability of the model recognition. Wang et al. [13] demonstrate Region Attention Network which is robust to posture and illumination, and Wang et al. [14] propose Self-Cure Network to make the model robust to uncertainty samples. Bargal et al. [15] extract features from VGG13, VGG16, and ResNet18, and then integrate all of them for classification. Zhang et al. [16] propose MSCNN network, which do expression recognition with cross-entropy loss to learn features with large between-expression varition and do face recognition by contrastive loss to reduce the varition in within expressions features.

B. ATTENTION MECHANISM

The attention mechanism originates from the study of human visual attention. Humans usually focus mainly on the important area instead of the whole to identify the category of the object. Recently, many studies have shown that integrating attention mechanism into the network to learn the correlation between the features can improve the effectiveness of the network, such as the Squeeze-and-Excitation(SE) module proposed by Hu [17], the SE module enables the model to selectively emphasize the features of important channels and suppress the features of unimportant channels through global information, thus improving the performance of the model. Woo et al. [18] proposed a CBAM module, which is trained by combining spatial attention and channel attention. First,
find the channel weights of the input features, let the input features multiply by the weights to obtain the feature map after channel attention, and then seek the spatial weight of this feature map, and let the features multiplied by the weights to obtain the features map after spatial attention. Then find the spatial weight of this feature map and let the feature multiply by which to get the feature map after spatial attention, so that the model can focus on both the important spatial and channel features, thus improving the model performance.

Closer to our work, Farzaneh et al. [19] take the view that when center loss is used to punish all features equally, there may be a distinction between important and unimportant features, from which they propose a deep attentive center loss that introduces an attention mechanism when penalizing features, which allows the model to learn the importance of each feature to highlight important features and suppress unimportant ones, which can further enhance the role of sharing affinity features.

III. METHOD
In this section, we first present an overview of the RASN and then describe the sharing affinity features module(Sharing-module) and the affinity features attention module(Attention-module). Finally, we introduce learning strategy and loss.

A. OVERVIEW
Since ResNet18 [7] overcomes the problem of vanishing gradient and can converge quickly, our RASN takes the classic ResNet18 as the backbone network, four Sharing-modules and Attention-modules are added after each of the four residual extraction layers (from Resnet layer1 to Resnet layer4). And the specific structure diagram is illustrated in Figure 2. Sharing-module obtains affinity features by averaging original features. The affinity features are shared with original features to compensate for the inadequate features learning of minority class expressions due to the insufficient number. Attention-module learns a weight for each channel of affinity features to obtain the affinity features’ importance. The learned weights are multiplied by affinity features to highlight expression-related affinity features and suppress expression-unrelated ones, which can further enhance the role of sharing affinity features.

B. SHARING AFFINITY FEATURES
Shi et al. [11] propose an Amend Representation Module (ARM) to enhance the model by Sharing Affinity block and De-albino block, since FER can be regarded as a face-specific classification task, there is a fact that different categories of expressions have affinity features, and sharing affinity features can improve the recognition accuracy of the model. However, since the way to acquire and share affinity features is used to assist De-albino block in ARM, its recognition accuracy is only minimally improved when using the Sharing Affinity block alone. In order to make full use of the affinity features among various expressions to address the sample imbalance, we propose a new method for sharing affinity features, which is different from ARM in the way of obtaining and sharing affinity features. To better illustrate the mechanism of sharing affinity features module to abate the influence of sample imbalance, an illustrative graph of which is shown in Figure 3.
As shown in Figure 3. The samples of sad and fear are simultaneously input into the convolutional neural network (CNN) to learn sad features and fear features (assuming sad is the minority class expression and fear is the majority class expression). As can be seen from the left side of the figure, the original learning strategy is that different classes expressions promote their corresponding feature learning, the learning routes of which are unrelated to each other. Since the number of sad samples is small and the fear samples is large, which will result in a good recognition of fear expressions and poor recognition of sad expressions. On the right side of the figure, it can be see that the learning routes of both two samples have been changed, each of the routes is augmented with a new learning route to the other, so the two routes are related to each other. This enables different expressions to learn from each other during expression feature extraction, which makes it possible for the minority classes to benefit from the majority classes and compensate for the inadequate features learning of minority classes due to insufficient numbers.

The specific implementations of our method are as follows, first, input original feature map \( F \) to obtain the feature map \( H = \times W \times C \times H \times W \) \((B\) represents the number of batch samples, \( C \) the channel size of the feature map, \( H \) the height of the feature map, \( W \) is the width of the feature map) to obtain affinity features \( F^{1 \times C \times H \times W} \) of the batch of samples, as shown in equation 1,

\[
F_g = \frac{1}{N} \sum_{i=0}^{N} F_{g}^{i \times C \times H \times W}
\]

where \( F_{g}^{i \times C \times H \times W} \) stands for the original feature of i-th image and \( N \) is the batch size.

Sharing affinity features means adding the affinity features \( F_g \) and the original input features \( F_s \) element-wisely, so that the model takes the affinity features into account when extracting features, as shown in equation 2,

\[
F_{gs} = F_g \oplus \lambda F_s,
\]

where \( \oplus \) denotes element-wise addition, \( F_g \) is the original input features, \( F_s \) is the affinity features, and \( \lambda \) is the adjustable hyper parameter, which reflects the contribution degree of the affinity feature \( F_g \) to the final feature \( F_{gs} \), the higher the value, the greater the contribution degree. Here, the first feature extraction layer (Resnet layer 1) of ResNet18 is used as an example. First, the original input features \( F_g \) extracted by Resnet layer 1 are input to a Sharing-module to obtain \( F_g \), which are multiplied by the hyperparameter \( \lambda \), and then add to the original input features \( F_s \) element-wisely to obtain the feature map \( F_{gs} \), and finally the feature map \( F_{gs} \) is used to Resnet layer 2 to achieve sharing affinity features. The afterward 3 feature extraction layers, from the second to the fourth layers uses the same way as the above-mentioned Resnet layer 1 to share the affinity features. In this way, all of the 4 feature extract layers will take the affinity features into consideration when extracting features, which can enhance the model's representation learning ability and the robustness of the sample imbalance, and thus raising the recognition accuracy.

**C. AFFINITY FEATURES ATTENTION**

By sharing affinity features, it is possible that minority classes can learn features from majority classes, so as to increase the robustness of the model to sample imbalance. However, there is reason to believe that the importance of each feature of affinity features is not the same. As with some expression-unrelated features, such as skin color, gender, and so on., there is no doubt that it will adversely affect the effect of sharing affinity features when the expression-unrelated features are given the same weight as those expression-unrelated ones. Therefore, an affinity features attention module is designed to obtain the weights of each channel of affinity features, and its network structure diagram is shown in Figure 4.
In order to obtain the weights of each channel of affinity features, a network layer is designed to learn the importance of each channel. The affinity features are input into this network layer to obtain the importance of each channel \( f_i \), as shown in Equation 3,

\[
f_i = W_2 \ast (\text{Relu}(W_1 \ast (\text{Flatten} \ast \text{BN}(F_s^{1 \times H \times W})))
\]

where \( F_s^{1 \times H \times W} \) represents the i-th channel feature map of \( F_s \), BN represents Batchnorm which normalizes \( F_s^{1 \times H \times W} \), Flatten indicates flattens \( F_s^{1 \times H \times W} \) to \( F_s^{1 \times HW} \), \( W_1 \) and \( W_2 \) represent the weights for the j-th fully connected (FC) layers where \( j = 1, 2 \). The two FC layers are inserted with Rectified Linear Units Relu to find the relationship between \( F_s^{1 \times H \times W} \) and \( f_i \).

After obtaining the importance \( f_i \) for each channel, the importance \( f_i \) is mapped into the interval (0, 1) by the sigmoid function for the weight \( a_i \) as shown in Equation 4,

\[
a_i = \frac{1}{1 + e^{-f_i}}
\]

where \( f_i \) is the importance of each channel and \( \frac{1}{1 + e^{-f_i}} \) is the sigmoid function with the value domain of (0, 1), by which can map \( f_i \) into (0, 1).

Finally, we get the attentive affinity features \( F_{as} \) through the \( a_i \) and \( F_s \) obtained above, as shown in Equation 5,

\[
F_{as} = A \otimes F_s.
\]

where \( A \) denotes a set of weights \( a_i \) for each channels and \( \otimes \) represents element-wise multiplication.

After obtaining the final attentive affinity features of \( F_{as} \), the sharing affinity features can be achieved by Equation 6,

\[
F_{gs} = F_g \oplus \lambda F_{as}.
\]

where \( \oplus \) denotes element-wise addition, \( F_g \) is the original input features, \( F_{as} \) is the attentive affinity features, and \( \lambda \) is the adjustable hyper parameter.

**D. LEARNING STRATEGY AND LOSS**

The gradient descent method is adopted to optimize the RASN. There are two main parameters for optimization, one for the CNN and another for the Attention-module, as shown in Algorithm 1. First, input a batch of samples \( D_m = \{(x_i, y_i) | i = 1, \ldots, m\} \), extract the original features and the attentive affinity features through the RASN to get the prediction result of the batch, and then cross-entropy loss is employed to calculate the loss between the prediction result and the sample labels,

\[
\text{Loss} = -\frac{1}{m} \sum_{i=1}^{m} p(y_i) \log q(x_i)
\]

where \( p(y_i) \) is the label corresponding to \( x_i \), \( q(x_i) \) is the prediction result of sample \( x_i \), and \( m \) is the number of samples in the batch.

**Algorithm 1 Learning Strategy of RASN**

**Require:** Training dataset(D), number of training(Maxepoch), learning rate(\( \alpha \)), number of iterations(num_iters)

**Ensure:** CNN parameters(\( \theta_c \)) and Attention-module parameters(\( \theta_a \))

1: while \( i < \text{Maxepoch} \) do
2: \hspace{10mm} for \( n=1 \) to \( \text{num_iters} \) do
3: \hspace{20mm} Sample a batch \( D_m = \{(x_i, y_i) | i = 1, \ldots, m\} \);
4: \hspace{20mm} Compute the original features \( F_g \) by CNN;
5: \hspace{20mm} Compute the attentive affinity features \( F_{as} \) by \( \text{Eq.1,3,4,5} \);
6: \hspace{20mm} Compute the prediction \( q(x_i) \) by \( F_g \) and \( F_{as} \);
7: \hspace{20mm} Compute the cross-entropy loss by \( \text{Eq.7} \):
8: \hspace{20mm} Compute the gradient of CNN by \( \text{Eq.8} \):
9: \hspace{20mm} Compute the gradient of Attention-module by \( \text{Eq.10} \):
10: \hspace{20mm} Updated the CNN parameters by \( \text{Eq.9} \):
11: \hspace{10mm} Updated the Attention-module parameter by \( \text{Eq.11} \):
12: \hspace{20mm} \( \theta_a^{n+1} \leftarrow \theta_a^n + \alpha g_a \);
13: \hspace{10mm} \( \theta_c^{n+1} \leftarrow \theta_c^n + \alpha g_c \);
14: \hspace{10mm} \( i \leftarrow i + 1 \);
15: \end{while}

The gradient \( g_c \) of the CNN is obtained by finding out the partial derivative of \( \theta_c \) for loss, and the parameter \( \theta_c \) for the CNN is optimized using gradient descent as follows,

\[
\frac{\partial \text{Loss}}{\partial \theta_c} = -\frac{1}{m} \sum_{i=1}^{m} p(y_i) \frac{\partial \log q(x_i)}{\partial \theta_c}
\]

\[
\theta_c^{n+1} = \theta_c^n + \alpha g_c
\]

where \( \theta_c^{n+1} \) is the updated CNN parameters, \( \theta_c^n \) is the pre-updated CNN parameters, \( g_c \) is the gradient of the CNN, and \( \alpha \) is the learning rate.

Finally, the gradient \( g_a \) of the Attention-module is obtained by taking the partial derivative of \( \theta_a \) for loss, and the parameter \( \theta_a \) for the Attention-module is optimized using gradient descent as follows,

\[
\frac{\partial \text{Loss}}{\partial \theta_a} = -\frac{1}{m} \sum_{i=1}^{m} p(y_i) \frac{\partial \log q(x_i)}{\partial \theta_a}
\]

\[
\theta_a^{n+1} = \theta_a^n + \alpha g_a
\]

where \( \theta_a^{n+1} \) is the updated Attention-module parameter, \( \theta_a^n \) is the Attention-module parameter before the update, \( g_a \) is the gradient of the Attention-module, and \( \alpha \) is the learning rate.
IV. EXPERIMENTS

In this section, we first introduce three public datasets. Then we demonstrate the experimental results of our RASN on the three datasets.

A. DATASETS

CK+ [20] is an extension of the Cohn-Kanade dataset, released in 2010, with 593 dynamic sequences of 123 objects and a total of 981 photos with labeled expressions, there are 7 categories of expressions, which are anger, disgust, fear, happiness, sadness, surprise, and contempt. In our work, 882 samples are selected as the training set and the remaining 99 samples are the test set.

FER2013 [9] was proposed in the 2013 kaggle expression recognition competition and comprises 35886 face expression images, which have been classified into three groups: training set (28708), public validation set (3589), and private validation set (3589). Each image has the size of $48 \times 48$ grayscale ones, and there are 7 categories of expressions, namely angry, natural, disgusted, fearful, happy, sad, and surprised.

RAF-DB [8] was proposed by Li S et al. in 2017 and consists of 29,672 face expression images, which have been classified into two major classes: training set (12271) and validation set (3068). The size of each image is $48 \times 48$ color image and there are 7 categories of expressions which are the same as that of FER2013.

B. IMPLEMENTATION DETAILS

1) PRE-PROCESSING

For the CK+ dataset, the Haar-Cascade classifier is used for face detection to intercept the face part of the image, the size of the train set images is scaled to $256 \times 256$, randomly rotated horizontally, and finally cropped to $224 \times 224$. The test set images are directly scaled up to $256 \times 256$ and then cropped to $224 \times 224$. For the FER2013 dataset, the training set is randomly flipped up and down and rotated horizontally, and the image size is scaled to $224 \times 224$ for the test set directly. As with the RAF-DB dataset, the training set is randomly erased and the image size is scaled to $224 \times 224$, and the image size of the test set is directly scaled to $224 \times 224$.

2) TRAINING

We train our RASN in an end-to-end manner. Selecting ResNet18 as the backbone pretrained on ImageNet dataset. In order to comply with the feature extraction of ResNet18, the number of channels of the Attention-module is set to 64, 128, 256 and 512 from the first to the fourth layers, respectively, and the hyperparameter $\lambda$ for the Sharing-modules is set to 0.5. The RASN is optimized using the Adam optimizer, with the number of iterations set to 60, the initial learning rate set to 0.001, and the exponentially decaying learning rate tuned with a factor of 0.9. The small batch sizes of the three datasets CK+, FER2013, and RAF-DB are set to 16, 128, and 128. The RASN is implemented based on the PyTorch framework, and all experimental results are achieved on an NVIDIA GPU GeForce RTX 2080Ti with 12GB of graphics memory.

C. ABLATION EXPERIMENT

1) COMPONENT ANALYSIS

In order to verify the role of Sharing-module and Attention-module of RASN, an ablation experiment is conducted on the RAF-DB dataset, and the results are shown in Table 1. Several conclusions can be drawn from the table. First, the recognition accuracy of the model is greatly increased when Sharing-module is introduced, because the model takes the affinity features into account when extracting features, thus enhancing the robustness of the model against sample imbalance. Second, the model recognition accuracy remains essentially unchanged when the Attention-module is introduced. The main reason is that the Attention-module is only used to learn the weights of affinity features, and it is of no effect without Sharing-module. Third, the Attention-module can make the Sharing-module improve by 1.76% and 2.51% without pre-training and with pre-training, respectively. This is accomplished by assigning high weight to expression-related affinity features and low weight to expression-unrelated affinity features, as a result, the Attention-module enhances the role of important affinity features and suppresses the role of unimportant affinity features, and further enhances the sharing affinity features effect.

2) THE EFFECT OF $\lambda$ ON MODEL RECOGNITION ACCURACY

To explore the influence of the hyperparameter $\lambda$ on the recognition accuracy of RASN, a comparison experiment of RASN with different hyper parameters $\lambda$ on the RAF-DB dataset is conducted. The variation of the recognition accuracy with the $\lambda$ is shown in Figure 5. It can be observed...
from Figure 5 that When $\lambda$ is around 0.5, the recognition accuracy of RASN reaches the highest and the model works best. Too small a value of $\lambda$ means that the affinity features account for less of the total features and the model only learns a small number of affinity features, so the shared affinity features play a less role and the model recognition rate is not improved much. Too large a value of $\lambda$ indicates that the proportion of affinity features in the total features is large, and the model learns too many affinity features and far fewer original features, also resulting in a low recognition rate. When $\lambda$ fluctuates around 0.5, the recognition accuracy of RASN is close to the highest value with little fluctuation, so the choice of $\lambda=0.5$ in our work is reasonable.

### D. EVALUATION OF RASN ON IMBALANCED DATASETS

To show the robustness of RASN to sample imbalance, we explore the robustness of RASN to sample imbalance on three sample imbalanced datasets CK+, FER2013, and RAF-DB. Specifically, we use ResNet18 as the backbone and compare our RASN with baseline (traditional ResNet18 without Sharing-module and Attention-module) and a state-of-the-art method for sample imbalance, namely Class Balance Loss (CBLoss) [21]. Analyzing the precise, recall, and F1-score of various expressions of the three methods to verify the robustness of our RASN to sample imbalance. The experimental results on the three datasets of CK+, FER2013, and RAF-DB are shown in Table 2, Table 3, and Table 4, respectively.

As shown in Table 2, Table 3, and Table 4, our RASN improves the baseline by a large margin. On CK+ dataset, the precision, recall, and f1-score of our RASN on various expressions are higher than that of the baseline. For the minority class expressions on CK+: sad, our RASN outperforms the baseline by 27.27%, 11.11%, and 20% in precision, recall, and f1-score, respectively. On FER203 dataset, all the above-mentioned metrics of our RASN are higher than that of the baseline on all expressions except Fear. For the minority class expression on FER2013: contempt, our RASN performs the baseline by 0.82%, 3.64%, and 2.46% in precision, recall, and f1-score, respectively. On RAF-DB dataset, all the above-mentioned metrics of our RASN are higher than that of the baseline on all expressions except Fear. For the minority class expression on RAF-DB: anger, our RASN outperforms the baseline by 4.29%, 3.71%, and 4.02% in precision, recall, and f1-score, respectively. The CBLoss rebalances the loss using the effective number of per class samples, which can reduce the overfitting of model to the majority class samples caused by sample imbalance. Compared with the baseline, CBLoss also improve the precision, recall, and f1-score on the three datasets, but it is still lower than RASN. In addition, we find that the improvement of RASN on CK+ is much higher than that of FER2013. This may be caused by the following reasons. On the one hand, CK+ is collected in the laboratory, and the collected objects are required to have a correct posture and no occlusion when the photos are taken, which leads to the good sample quality of this dataset. On the other hand, FER2013 is automatically collected on the web with a large number of low-quality images and noisy labels. Therefore, our RASN can achieve greater improvement on datasets with higher sample quality.

### E. VISUALIZATION ANALYSIS

#### 1) ATTENTION VISUALIZATION

To better analyze the effectiveness of RASN, We randomly select different class samples from RAF-DB and then use Grad-CAM [22] to make attention visualization of the main areas of concern of RASN and baseline (traditional ResNet18 without Sharing-module and Attention-module). The experimental results are shown in Figure 6. From the figure, it is to be noted that for the surprise class expressions, RASN mainly focuses on the eyes and eyebrows. For the happy expressions, RASN mainly focuses on its mouth and the center of the face. For the anger expressions, RASN mainly focuses on its mouth and eyebrow area. These phenomena are consistent with reality and demonstrate that our RASN does learn the key features for discriminating expressions. We believe this may be because the Attention-module of RASN effectively learns the importance of affinity features.

| TABLE 2. The evaluation of RASN on CK+. |
|-----------------------------------------|
| Method       | Metric | Angry | Disgust | Fear | Happy | Sad | Surprise | Contempt |
|--------------|--------|-------|---------|------|-------|----|---------|----------|
| Baseline     | Precision | 85.71 | 81.82 | 100  | 100  | 72.73 | 100  | 80.00   |
| CBLoss [21]  | RASN   | 100  | 85.71 | 100  | 100  | 64.29 | 100  | 100     |
| CBLoss [21]  | Recall | 90.00 | 100  | 100  | 100  | 88.89 | 100  | 66.67   |
| Baseline     | F1-score | 63.16 | 90.00 | 100  | 100  | 80.00 | 100  | 72.73   |
| CBLoss [21]  | F1-score | 66.67 | 92.31 | 100  | 100  | 78.26 | 97.87 | 90.91   |
| Baseline     |             | 85.71 | 92.31 | 100  | 100  | 100   | 100  |         |

| TABLE 3. The evaluation of RASN on FER2013. |
|-----------------------------------------|
| Method       | Metric | Suprise | Fear | Disgust | Happy | Sad | Angry | Neutral |
|--------------|--------|---------|------|---------|-------|----|-------|---------|
| Baseline     | Precision | 84.23 | 77.78 | 64.38 | 94.66 | 83.71 | 83.57 | 83.62   |
| CBLoss [21]  | RASN   | 87.02 | 80.00 | 62.88 | 93.74 | 84.24 | 79.87 | 83.31   |
| CBLoss [21]  | Recall | 90.88 | 54.05 | 51.88 | 94.85 | 83.89 | 78.40 | 87.35   |
| Baseline     | F1-score | 89.67 | 68.92 | 68.75 | 96.88 | 83.89 | 75.93 | 97.21   |
| CBLoss [21]  | F1-score | 88.32 | 64.52 | 56.85 | 94.30 | 84.07 | 79.13 | 85.28   |
| Baseline     |             | 89.94 | 71.83 | 73.58 | 96.07 | 87.46 | 81.46 | 92.46   |

| TABLE 4. The evaluation of RASN on RAF-DB. |
|-----------------------------------------|
| Method       | Metric | Suprise | Fear | Disgust | Happy | Sad | Angry | Neutral |
|--------------|--------|---------|------|---------|-------|----|-------|---------|
| Baseline     | Precision | 84.23 | 77.78 | 64.38 | 94.66 | 83.71 | 83.57 | 83.62   |
| CBLoss [21]  | RASN   | 87.02 | 80.00 | 62.88 | 93.74 | 84.24 | 79.87 | 83.31   |
| CBLoss [21]  | Recall | 90.88 | 54.05 | 51.88 | 94.85 | 83.89 | 78.40 | 87.35   |
| Baseline     | F1-score | 89.67 | 68.92 | 68.75 | 96.88 | 83.89 | 75.93 | 97.21   |
| CBLoss [21]  | F1-score | 88.32 | 64.52 | 56.85 | 94.30 | 84.07 | 79.13 | 85.28   |
| Baseline     |             | 89.94 | 71.83 | 73.58 | 96.07 | 87.46 | 81.46 | 92.46   |
Sharing the attentive affinity features makes the RASN learn the key features for predicting expressions, which enables RASN to focus on important regions when discriminating expressions and makes the prediction results more accurate.

2) VISUALIZATION OF LEARNED FEATURES

We use t-SNE [23] to visualize the features learned by the baseline (traditional ResNet18 without Sharing-module and Attention-module) and our RASN to demonstrate the effectiveness of RASN. The initialization method of t-SNE is set to PCA initialization, and the embedding space dimension is set to 2. The experimental results are shown in Figure 7. It turns out that the features learned by RASN are more discriminative compared to baseline, with the larger inter-class distance and the smaller intra-class distance. And the minority class expressions such as surprise and fear clearly form a clear boundary with other expressions. It could be believed that this may be because sharing attentive affinity features enables RASN to learn features from majority class samples when learning features for minority class expressions, thus enhancing the recognition ability of the model for minority class expressions and making the minority class expressions better distinguishable from other classes of expressions.

F. COMPARISON WITH STATE-OF-THE-ART METHODS

Although much effort has been invested in solving the sample imbalance problem, our RASN still achieves better recognition accuracy compared to other state-of-the-art methods and baseline (traditional ResNet18 without Sharing-module and Attention-module). Tables 5, 6, and 7 show the test accuracy comparison on CK+, FER2013, and RAF-DB, respectively.

As can be seen from the table, for CK+, the second-highest recognition accuracy of the test set is Pre-train CNN, which uses transfer learning technique to overcome the shortage of training samples. For FER2013, the second-highest recognition accuracy of the test set is SAP [34], which is a sample awareness-based expression recognition method, in which a Bayesian classifier is used to select the most appropriate classifier from a set of candidate classifiers for the current test sample, and then the classifier is used to perform expression recognition on the current sample. For RAF-DB, the test set with the second-highest recognition accuracy is DACL [19], which combines center loss and an attention mechanism to selectively penalize features for enhanced discrimination.

Compared with the above methods, our proposed RASN outperforms those state-of-the-art methods, achieving 96.97%, 71.44%, and 90.91% on CK+, FER2013, RAF-DB, respectively. It has been proved that the RASN performs well on the expression recognition task and has good generalization ability. It is to be noted that our RASN has higher accuracy on CK+ and lower accuracy on RAF-DB and FER2013. This is mainly caused by that CK+ is collected in the laboratory, while RAF-DB and FER2013 are collected from real scenes. The CK+ collected in the laboratory has a good posture, no occlusion, and higher label reliability. However, RAF-DB and FER2013 collected in the wild have problems such as low sample quality and noisy labels.
Especially on FER2013, most of the labels are automatically annotated, resulting in low label reliability.

To confirm the accuracy of our RASN, we randomly demonstrate 10 samples from CK+, FER2013, and RAF-DB. As shown in Figure 8, our RASN has only one wrong prediction sample on CK+, RAF-DB, and three wrong prediction samples on FER2013. This is consistent with the fact that our RASN achieves 96.97%, 71.44%, and 90.91% accuracies on CK+, FER2013, and RAF-DB, respectively.

In order to evaluate the recognition effect of RASN on different types of expressions on the three datasets of CK+, FER2013, and RAF-DB, the confusion matrices of RASN on the three datasets are shown in Figure 9. It can be seen from the figure that RASN has a higher recognition accuracy of various expressions on CK+. Except for the angry expression recognition accuracy, which is 75%, the remaining expression recognition accuracy is 100%. On FER2013, three types of expressions of fear, sadness, and anger have low recognition accuracy, while the rest of the expressions perform well. On RAF-DB, except for the recognitions of fear and disgust, of which the recognition accuracy is low, the recognition accuracy of the other five categories of expressions reaches as high as 75%, especially the happy and neutral expressions reach 97%.

G. MODEL EFFICIENCY

Our RASN can also be used as a real-time expression recognition method. To demonstrate the efficiency of our RASN, we compare RASN with baseline methods (traditional ResNet18 without Sharing-module and Attention-module) in terms of the number of parameter, the number of float-point operation, and the frame rate. As shown in Table 8, in terms of computing power, compared with Baseline, the number of parameters of RASN has increased from 11.2M to 14.4M, and the FLOPs of RASN have increased from
TABLE 8. The parameter amount, floating-point operands, and frame rate of RASN and Baseline.

| Method   | Parameters | PLOPs | FPS |
|----------|------------|-------|-----|
| Baseline | 11180103   | 18235253576 | 245 |
| RASN     | 144114147  | 1827511040  | 179 |

1,823,525K to 1,827,511K. The FPS of RASN has decreased from 245 to 179. Although the RASN increases the computational cost and reduces the FPS, it can be seen from Tables 5, 6, and 7 that the RASN improves accuracy by about 3%.

V. CONCLUSION

Facial expression recognition plays an important role in human daily life. However, the sample imbalance in expression datasets will lead to the overfitting of model to majority classes expressions. This will cause the model to have poor recognition accuracy for minority classes.

In this paper, based on the fact that different expressions have affinity features, which makes it possible for the minority classes to benefit from the majority classes in the expression feature extraction process, we propose a Residual Attentive Sharing Network (RASN) to solve the problem of sample imbalance. RASN is formed by adding a Sharing-module and an Attention-module to each of the four feature extraction layers (from ResNet layer1 to ResNet layer4) of ResNet18. The Sharing-module shares affinity features containing features of minority classes and majority classes to compensate for the inadequate feature learning of minority classes due to the insufficient number. And the Attention-module learns the importance of each channel of affinity features to highlight expression-related affinity features and suppress expression-unrelated ones, which can further enhance the role of sharing affinity features. The RASN is evaluated on three publicly available databases, CK+, FER2013, and RAF-DB. Experimental results demonstrate that our RASN achieved superior results and was robust to sample imbalance.

In the future work, we will continue to study how to build a more effective RASN to improve the robustness to sample imbalance. In addition, we will be testing our method in different networks.

REFERENCES

[1] G. Levi and T. Hassner, “Emotion recognition in the wild via convolutional neural networks and mapped binary patterns,” in Proc. ACM Int. Conf. Multimodal Interact., Nov. 2015, pp. 503–510.

[2] S. Berretti, A. D. Bimbo, P. Pala, B. B. Amor, and M. Daoidi, “A set of selected SIFT features for 3D facial expression recognition,” in Proc. 20th Int. Conf. Pattern Recognit., Aug. 2010, pp. 4125–4128.

[3] N. Zeng, H. Zhang, B. Song, W. Liu, Y. Li, and A. M. Dobaie, “Facial expression recognition via learning deep sparse autoencoders,” Neurocomputing, vol. 273, pp. 643–649, Jan. 2018.

[4] S. Liu, Y. Tian, and C. Wan, “Facial expression recognition method based on Gabor multi-orientation features fusion and block histogram,” Acta Automatica Sinica, vol. 37, no. 12, pp. 1455–1463, 2011.

[5] V. Kumar, S. Rao, and L. Yu, “Noisy student training using body language dataset improves facial expression recognition,” in Proc. Eur. Conf. Comput. Vis. Cham, Switzerland: Springer, 2020, pp. 756–773.

[6] I. Bendjoudi, F. Vanderhaegen, D. Hamad, and F. Dornaika, “Multi-label, multi-task CNN approach for context-based emotion recognition,” Inf. Fusion, vol. 76, pp. 422–428, Dec. 2021.

[7] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2016, pp. 770–778.

[8] S. Li, W. Deng, and J. Du, “Reliable crowdsourcing and deep locality-preserving learning for expression recognition in the wild,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jul. 2017, pp. 2852–2861.

[9] I. J. Goodfellow et al., “Challenges in representation learning: A report on three machine learning contests,” in Proc. Int. Conf. Neural Inf. Process. Springer, 2013, pp. 117–124.

[10] S. Xie, H. Hu, and Y. Chen, “Facial expression recognition with two-branch disentangled generative adversarial network,” IEEE Trans. Circuits Syst. Video Technol., vol. 31, no. 6, pp. 2359–2371, Jun. 2021.

[11] J. Shi, S. Zhu, and Z. Liang, “Learning to amend facial expression representation via de-albino and affinity,” 2021, arXiv:2103.10189.

[12] D. Meng, X. Peng, K. Wang, and Y. Qiao, “Frame attention networks for facial expression recognition in videos,” in Proc. IEEE Int. Conf. Image Process. (ICIP), Sep. 2019, pp. 3866–3870.

[13] K. Wang, X. Peng, J. Yang, D. Meng, and Y. Qiao, “Region attention networks for pose and occlusion robust facial expression recognition,” IEEE Trans. Image Process., vol. 29, pp. 4057–4060, 2020.

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

[15] S. A. Bargal, E. Barsoum, C. C. Ferrer, and C. Zhang, “Emotion recognition in the wild from videos using images,” in Proc. 18th ACM Int. Conf. Multimodal Interact., Oct. 2016, pp. 433–436.

[16] K. Zhang, Y. Huang, Y. Du, and L. Wang, “Facial expression recognition based on deep evolutionary spatial-temporal networks,” IEEE Trans. Image Process., vol. 26, no. 9, pp. 4193–4203, Sep. 2017.

[17] J. Hu, L. Shen, and G. Sun, “Squeeze-and-excitation networks,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Jun. 2018, pp. 7132–7141.

[18] S. Woo, J. Park, Y.-J. Lee, and I. S. Kweon, “CBAM: Convolutional block attention module,” in Proc. Eur. Conf. Comput. Vis. (ECCV), 2018, pp. 3–19.

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

[20] P. Lucey, J. F. Cohn, T. Kanade, J. Saragih, Z. Ambadar, and I. Matthews, “The extended Cohn-Kanade dataset (CK+): A complete dataset for action unit and emotion-specified expression,” in Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.-Workshops, Jun. 2010, pp. 94–101.

[21] Y. Cui, M. Jia, T.-Y. Lin, Y. Song, and S. Belongie, “Class-balanced loss based on effective number of samples,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2019, pp. 9268–9277.

[22] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, “Grad-CAM: Visual explanations from deep networks via gradient-based localization,” in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), Oct. 2017, pp. 619–626.

[23] L. Van der Maaten and G. Hinton, “Visualizing data using t-SNE,” J. Mach. Learn. Res., vol. 9, no. 11, pp. 1–27, 2008.

[24] M. Liu, S. Li, S. Shan, and X. Chen, “AU-inspired deep networks for facial expression feature learning,” Neurocomputing, vol. 159, pp. 126–136, Jul. 2015.

[25] J. Cai, Z. Meng, A. S. Khan, Z. Li, J. O’Reilly, and Y. Tong, “Island loss for learning discriminative features in facial expression recognition,” in Proc. 13th IEEE Int. Conf. Autom. Face Gesture Recognit. (FG), May 2018, pp. 302–309.

[26] S. Xie, H. Hu, and Y. Wu, “Deep multi-path convolutional neural network joint with salient region attention for facial expression recognition,” Pattern Recognit., vol. 92, pp. 177–191, Aug. 2019.

[27] J. Shao and Y. Qian, “Three convolutional neural network models for facial expression recognition in the wild,” Neurocomputing, vol. 355, pp. 82–92, Aug. 2019.

[28] J. Shao and Q. Cheng, “E-FCNN for tiny facial expression recognition,” Int. J. Speech Technol., vol. 51, no. 1, pp. 549–559, Jan. 2021.

[29] C. Liu, K. Hirota, J. Ma, Z. Jia, and Y. Dai, “Facial expression recognition using hybrid features of pixel and geometry,” IEEE Access, vol. 9, pp. 18876–18889, 2021.
[30] A. Mollahosseini, D. Chan, and M. H. Mahoor, “Going deeper in facial expression recognition using deep neural networks,” in Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV), Mar. 2016, pp. 1–10.

[31] Z. Qin and J. Wu, “Visual saliency maps can apply to facial expression recognition,” 2018, arXiv:1811.04544.

[32] P. Giannopoulos, I. Perikos, and I. Hatziilygeroudis, “Deep learning approaches for facial emotion recognition: A case study on FER-2013,” in Advances in Hybridization of Intelligent Methods. Springer, 2018, pp. 1–16.

[33] M. Georgescu, R. T. Ionescu, and M. Popescu, “Local learning with deep and handcrafted features for facial expression recognition,” IEEE Access, vol. 7, pp. 64827–64836, 2019.

[34] H. Li and G. Wen, “Sample awareness-based personalized facial expression recognition,” Appl. Intell., vol. 49, no. 8, pp. 2956–2969, 2019.

[35] S. Minaei, M. Minaei, and A. Abdolrashidi, “Deep-emotion: Facial expression recognition using attentional convolutional network,” Sensors, vol. 21, no. 9, p. 3046, 2021.

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

[37] J. Cui, Z. Meng, A. Shehab Khan, Z. Li, J. O’Reilly, and Y. Tong, “Probabilistic attribute tree in convolutional neural networks for facial expression recognition,” 2018, arXiv:1812.07067.

[38] S. Saurav, R. Saini, and S. Singh, “EmNet: A deep integrated convolutional neural network for facial emotion recognition in the wild,” Appl. Intell., vol. 51, no. 8, pp. 5543–5570, 2021.

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

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

JIAHONG YANG received the Ph.D. degree from the School of Information Science and Engineering, Central South University, Changsha, China. He is currently a Full Professor with the College of Information Science and Engineering, Hunan Normal University, Changsha. He has published several academic articles in well-known international journals, including Journal of Systems and Control Engineering and IEEE TRANSACTIONS ON NanoBioscience. His research interests include deep learning, image processing, and computer vision.

ZHISHENG LV was born in Hengyang, Hunan, China, in 1999. He received the B.S. degree from the Hunan University of Arts and Science, Changde, China, in 2020. He is currently pursuing the M.S. degree with the School of Information Science and Engineering, Hunan Normal University, Changsha, China. His research interests include deep learning and image processing.

SEN YANG was born in Zhuzhou, Hunan, China, in 1997. He received the B.S. degree from Hubei Polytechnic University, Jingzhou, China, in 2019. He is currently pursuing the M.S. degree with the School of Information Science and Engineering, Hunan Normal University, Changsha, China. His research interests include deep learning and image processing.

KAI KUANG was born in Shaoyang, Hunan, China, in 1998. He received the B.S. degree from Hunan Normal University, Changsha, China, in 2020, where he is currently pursuing the M.S. degree with the School of Information Science and Engineering. His research interests include deep learning and image processing.

LIUMING XIAO received the B.E. degree from Hunan Normal University and the M.S degree from Central South University. She is currently working as a Teacher at Hunan Normal University. Her research interests include deep learning and image processing.

QIANG TANG received the B.E. and M.S. degrees from Hunan Normal University. He is currently working as a Teacher at Hunan Normal University. His research interests include deep learning and image processing.