Mimicking the Annotation Process for Recognizing the Micro Expressions

Bo-Kai Ruan
National Yang Ming Chiao Tung University
justin.ee08@nycu.edu.tw

Ling Lo
National Yang Ming Chiao Tung University
linglo.ee08@nycu.edu.tw

Hong-Han Shuai
National Yang Ming Chiao Tung University
hhshuai@nycu.edu.tw

Wen-Huang Cheng
National Yang Ming Chiao Tung University
whcheng@nycu.edu.tw

ABSTRACT

Micro-expression recognition (MER) has recently become a popular research topic due to its wide applications, *e.g.*, movie rating and recognizing the neurological disorder. By virtue of deep learning techniques, the performance of MER has been significantly improved and reached unprecedented results. This paper proposes a novel architecture to mimic how the expressions are annotated. Specifically, during the annotation process in several datasets, the AU labels are first obtained with FACS, and the expression labels are then decided based on the combinations of the AU labels. Meanwhile, these AU labels describe either the eyes or mouth movements (mutually-exclusive). Following this idea, we design a dual-branch structure with a new augmentation method to separately capture the eyes and mouth features and teach the model what the general expressions should be. Moreover, to adaptively fuse the area features for different expressions, we propose Area Weighted Module to assign different weights to each region. Additionally, we set up an auxiliary task to align the AU similarity scores to help our model capture facial patterns further with AU labels. The proposed approach outperforms other state-of-the-art methods in terms of accuracy on the CASME II and SAMM datasets. Moreover, we provide a new visualization approach to show the relationship between the facial regions and AU features.

CCS CONCEPTS

• **Computing methodologies** → **Computer vision**; **Image representations**.

KEYWORDS

micro-expression recognition; AU-feature learning

ACM Reference Format:
Bo-Kai Ruan, Ling Lo, Hong-Han Shuai, and Wen-Huang Cheng. 2022. Mimicking the Annotation Process for Recognizing the Micro Expressions. In *Proceedings of the 30th ACM International Conference on Multimedia (MM '22), October 10–14, 2022, Lisboa, Portugal.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3503161.3548185

MM '22, October 10–14, 2022, Lisboa, Portugal

ACM ISBN 978-1-4503-9203-7/22/10…$15.00
https://doi.org/10.1145/3503161.3548185

1 INTRODUCTION

Unlike facial expressions, micro expressions cannot be fabricated and thus are able to reveal the true emotions of a person, which is essential in emotional tasks, *e.g.*, movie rating [25], deception detection [23], and genuine emotions capturing and hiding [9]. Although recognizing the micro expression is valuable, the task is challenging since the period of a micro expression lasts only 500ms [39], which is approximately the time for a person to blink.

As such, several preprocessing methods have been proposed to facilitate the recognition. Specifically, each micro-expression sample is a sequence of frames, where the first frame of a micro expression is annotated as the *onset* frame, and the climax frame of an expression is labeled as the *apex* frame, and the end frame of an expression is marked as the *offset* frame. These frames can be regarded as the key to extracting useful features. A commonly-used approach is to create a magnified frame for recognition by an *onset* frame and an *apex* frame with motion magnification techniques. These magnification techniques include Eulerian Motion Magnification (EMM) [32] and Learning-Based Motion Magnification (LBMM) [22]. Another useful feature is Action Unit (AU), which describes the movement of facial muscles. Since facial expressions are formed through facial muscles, AUs can be applied to extract the facial information or used for auxiliary tasks.

In addition, approaches to classifying the micro expressions can be roughly classified into handcrafted-based, CNN-based, and graph-based methods. On the one hand, handcrafted-based methods utilize the handcrafted features for classification. These handcrafted features include using Local Binary Patterns (LBP) to extract textures of an image on three orthogonal planes (LBP-TOP [41]) or using Tensor Independent Color Space (TICS [30]) to obtain spatial, temporal, and color information before recognizing the labels with LBP. Other approaches, such as histogram of gradients (HOG) [6] and histogram of optical flow (HOOF) [18], utilize the differences between two frames to observe the changed areas.

On the other hand, CNN-based approaches extract the features directly from an input. The extracted features from the model are sent into linear layers to predict the results. A standard practice to create a CNN model is to incorporate the dual-branch or even the tri-branch design, *i.e.*, using more than one branch to help a model learn variant features from multiple inputs. For instance, Khor et al. [10] propose using dual-branch truncated AlexNet [12] to extract two

different features created by an optical-flow frame, an optical-strain frame, and a gray *apex* frame. By contrast, TSCNN [26] uses three-stream convolutional networks to process three different features—an *apex* frame, an optical-flow frame, and the grid blocks of the *apex* frame. Additionally, STRCN-G [35] incorporates Recurrent Convolutional Networks, which are used to extend the scale of receptive fields, to extract the patterns from an optical-flow input.

On top of that, graph-based methods first extract the node embeddings or the image representations and enhance the node features from the neighboring nodes to predict the micro expressions. For example, MER-GCN [19] predicts the expressions by combining the image features extracted by a 3D-CNN net with the AU features obtained from the AU relations graph. Instead of training extra graph neural networks, Graph-TCN [15] regards facial landmarks as the nodes and uses the shape representations in LBMM as the node features. Afterward, the graph features are sent into adapted temporal convolutional networks [1] to extract points and edges information. Following the same preprocessing in Graph-TCN, FMER [14] create two branches for model to 1) extract image features through transformer layers [4] and 2) learn the AU relations through graph convolutional networks.

Although the existing methods provide different learning strategies, they do not take into consideration how these expression labels are given. In fact, datasets can be annotated through the combination of AU labels [38], which are obtained from Facial Action Coding System (FACS) [5]. These AU labels either represents eyes or mouth movements but not both[1]. Additionally, using the whole faces for training is not appropriate because the eyes and mouth have different patterns, which should be captured by learning different filters separately. In order to address these issues, we design a novel architecture with dual-branch learning by separating an input image into eyes and mouth areas. Furthermore, to help model learn what the general expressions are, we propose an augmentation method named Region Replacing by replacing the eyes or mouth image with another same-class sample's counterpart to form diversified inputs. The two region inputs are then sent into the same base layer and two different branch layers to extract eyes and mouth features. Afterward, the extracted features are fed into the proposed Area Weighted Module (AWM) to obtain the area weights, which are used to determine the features importance because different expressions require different combinations. Finally, the region features are summed up to predict the micro expressions for simulating the AU combination stage.

In addition, we aim to merge the AU labels into our training to obtain more facial information. However, the existing methods may encounter the imbalanced AU problem by directly utilizing the AU labels, which can undermine the whole training process. To include AU labels into our training while preventing our model from the imbalanced problem, we create a new AU-based task called AU alignment to train our model on AU similarity scores by Weighted Supervised Contrastive Learning. Finally, we propose a new visualization approach called AU-CAM to observe how the AU features are captured by our model and which facial areas contribute most to the AU features. Our contributions are summarized below:

- We design a dual-branch architecture with AU alignment for micro-expression recognition based on how humans annotate the expressions by separating the eyes and mouth regions at the beginning and fusing the features in the end.
- To better understand how our model captures the facial features, we design AU-CAM to visualize how our model connects the AU features with the facial regions.

2 RELATED WORK

2.1 Micro-Expression Recognition

Micro-Expression Recognition is first proposed using the original image patterns (handcrafted features) such as LBP-TOP [41], LBP-SIP [31], TICS [30], optical flow [6, 18], and HOOF [18] to predict the expressions. Many researchers now turn to deep-learning approaches, universally acknowledged as the powerful methods in feature extraction. These approaches include using CNN architectures [10, 26, 28, 35] to find the essential patterns or building graph neural networks [13–15, 19, 36] by utilizing the provided AU labels in datasets to capture the subject's facial motions. The abovementioned approaches, however, do not consider how the micro expressions are annotated in the real world. Our proposed approach simulates the annotation process by separating an input image into two facial regions at the beginning and fusing the weighted region features at the end.

2.2 Contrastive Learning

Contrastive learning (CL) is a subset of self-supervised learning, which can be trained without annotated labels but can use data itself as the ground truth. The core concept of CL is to attract the same-class labels to cluster together and repel others away. The distance between two samples is examined based on an similarity score with the InfoNCE loss function [27]. Recently, more and more research works have focused on solving different applications with contrastive learning, *e.g.*, AU prediction [37], multiple object tracking [7], and music representation [43]. Additionally, based on the design of contrastive learning, Khosla et al. [11] propose Supervised Contrastive Learning (SCL) by combining the ground-truth labels into the objective function, achieving higher performance than all the past supervised learning. Inspired by SCL, we develop a new approach called Weighted SCL by providing different weights in the objective function to include distance information to align the AU similarly without facing the imbalanced AU problem.

2.3 Model Interpretability

Model interpretability help researchers know how a model generates the results. Hence, we can figure out whether a model makes the right decision based on reasonable information. An early visualized approach called Saliency Map measures the gradient of each pixel to the result to know how strong a pixel can influence the outcome. Recently, Class Activation Map (CAM) [42] has provided an alternative visualization by computing the weighted sum of the outputs in the last CNN layer. Other CAM-based approaches [2, 24, 29] are proposed to tackle the CAM issues and can be used outside the CNN structure. To visualize the regions that contribute most to the final AU features but not how a model determines the

[1]Only few of them, which are AU51∼58 that represent the head motions, include both eyes and mouth.

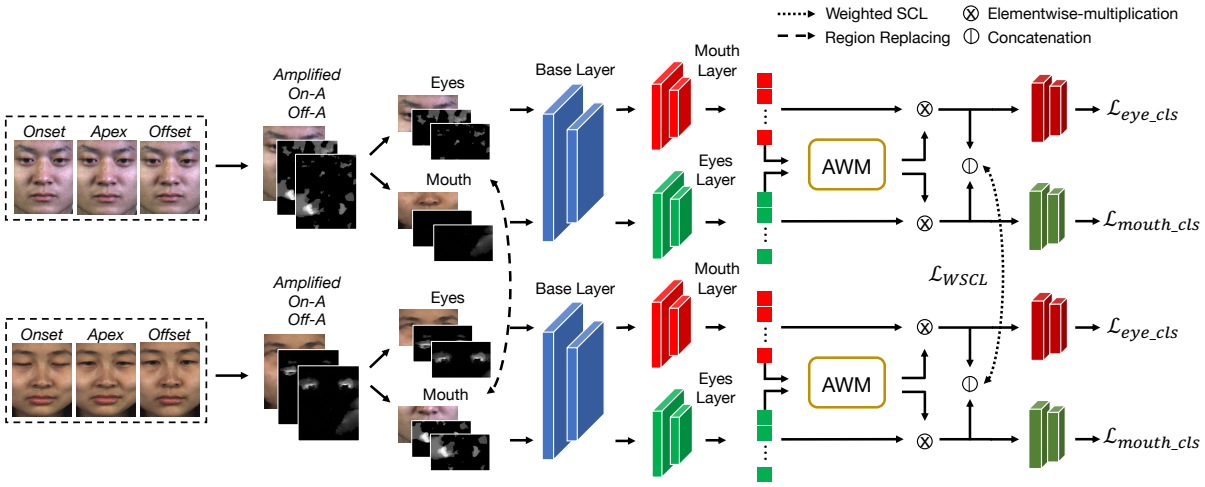

Figure 1: Our model architecture. The input images are first processed to obtain the input features and are separated into eyes and mouth images. Next, the Region Replacing is included to replace the original mouth image with another same-class sample's mouth image (*happiness* in this example). Afterward, the region images are sent into the same base layer and different branch layers to obtain region features. To fuse the features with different importance, AWM is used to obtain the weights. To further extract the facial information, AU alignment is applied on concatenated features provided by AWM.

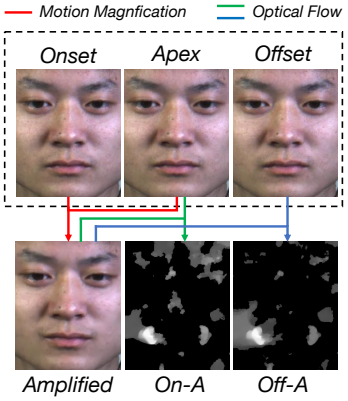

Figure 2: Illustration of the preprocessing. The amplified frame is first generated by applying motion magnification to the *onset* and the *apex* frames. Afterward, we create the *On-A* (*Off-A*) frame by computing the optical flow between the amplified frame and the *apex* (*offset*) frame.

final results, we construct AU-CAM based on Grad-CAM [24] to connect AU features and facial regions.

3 PROPOSED METHOD

Figure 1 illustrates the architecture of the proposed model. Specifically, an input image is first preprocessed to obtain a magnified motion and two optical-flow inputs. Next, these three inputs are concatenated and are separated into the eyes and mouth areas to follow the representations of AU labels. To avoid the spurious correlation and to augment the data, we randomly replace the same-class subjects' region images to form new inputs to teach our model recognize the general expressions. The input is then sent into the dual-branch architecture to extract high-level feature maps. Afterward, instead of directly combining the features from the two separated areas, we build Area Weighted Module (AWM) to learn the weights for adaptively fusing two kinds of features. Moreover, to ensure our model can extract useful facial features, we attach an auxiliary task based on AUs, namely AU alignment, to train our model further. Finally, the weighted features are summed up to generate predicted labels for simulating the combination of AU labels in the annotation process.

3.1 Feature Preprocessing

3.1.1 Input Features. Due to its hard-perceived changes, a raw image sequence cannot provide sufficient features. Motion magnification usually plays a vital role in data preprocessing to generate discriminative features and can transform the micro expressions into macro expressions. However, it is still challenging for a model to locate the changed areas across the frames from magnified motions. On the other hand, while the optical flow is commonly-used for its concise description of the movement of regions, it cannot provide clear patterns to enhance the slight movement of a face.

To address the issues mentioned above, we combine these two features since they complement each other. The combining features contain stronger patterns for our model to distinguish different micro expressions. Specifically, we adopt Learning-based Motion Magnification [22] to create amplified frames. Next, we create an *On-A* (*Off-A*) frame with the optical flow between an *onset* (*offset*) and an amplified frame. We also try getting the *On-A* and *Off-A* frames by replacing the amplified frame with the *apex* frame (discussed in Section 4.3). Finally, these three image frames-amplified frame, *On-A* frame, and *Off-A* frame are concatenated to form a new input feature. The detailed process is shown in Figure 2.

Table 1: Details of the backbone structure.

Usage	ResNet18 layers	Input shape	Output shape
Base layer	Layer 1	(3, 224, 224)	(64, 28, 28)
	Layer 2	(64, 28, 28)	(128, 14, 14)
Branch layer	Layer 3	(128, 14, 14)	(256, 7, 7)
	Layer 4	(256, 7, 7)	(512, 1)

3.1.2 Facial Separation. In this stage, an input image is separated based on two regions-eyes and mouth. The eyes image is the upper-half part of the image, and the rest half part is the mouth image. This step is to simulate that AU labels either describe the eyes or mouth movements but not both. Additionally, separating the images to fit a dual-branch architecture helps our model obtain diversified features due to different critical patterns of eyes and mouth images.

3.2 Region Replacing

Using a dual-branch architecture enables our model to learn to extract variant features. However, training a model to obtain the significant features requires considerable data, which could be limited in MER. Hence, we propose a new data augmentation technique called Region Replacing (RR) to replace one's region image with the region image of another sample in the same class to facilitate the learning of the general expressions.

The basic idea of RR is to replace a region image (eyes or mouth) with another same-class sample's region image to form a new input. To simplify the implementation process, we simply replace the original mouth image with another mouth image. This can be achieved by following steps. Firstly, for each preprocessed input sample, we randomly pick another sample with the same label in the dataset and take its mouth image. Afterward, we combine the original eyes image with the new mouth image to create a different input[2]. For instance, in Figure 1, both inputs are labeled with *happiness*, and both mouth images are replaced with each other's mouth image to form new inputs (we suppose both samples randomly select each other in this scenario).

It is worth noting that using RR, the bias between different subjects with the same class could be better eliminated. As such, the model can learn more general features from different input combinations without bringing external noise into our training since the same expression has similar facial movements.

3.3 Feature Extraction

After separating an image into two parts, each region image is first sent into the base layer and then to the branch layer. Both base and branch layer are part of the ResNet18 model [8]. Moreover, to encourage the model to extract diversified features, we include the Shuffle Attention Module [40], which separates the features into different groups and provides each feature within a group with different spatial and channel weights. However, these weights are shared between the groups, which can hinder our model to extract variant features because we force each group to use the same weights. Instead of sharing weights between the groups, we make our model learn different weights for different groups.

During the extraction stage, both eyes and mouth images are sent into the same base layer created by the first two ResNet18 layers to extract similar features. Later, the extracted features of the eyes and mouth areas are passed into different branch layers made by the last two ResNet18 layers. The details of the model structure are listed in Table 1.

[2]In the experiments, we apply RR with a 0.5 probability to randomly substitute the combined frames for the original frames.

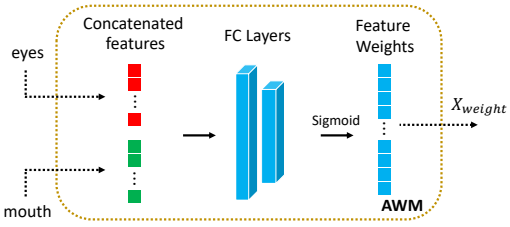

Figure 3: Illustration of Area Weighted Module. The eyes and mouth, in the beginning, represent the features obtained from each branch layer.

3.4 Merging the Region Features

To follow how the expressions are decided by the combination of AU labels, the features are fed into the Area Weighted Module (AWM) to provide each feature a different weight before predicting the labels. The structure of AWM is shown in Figure 3. Specifically, to obtain the weights for different area features, the outputs X_{eyes} and X_{mouth} from the two branches are concatenated and are fed into two linear layers. The features weight $X_{weight} \in \mathbb{R}^{1024}$ is computed by the *sigmoid* function of the last layer output:

$$X_{weight} = \text{sigmoid}(W_2 f(W_1[X_{eyes}|X_{mouth}] + b_1) + b_2) \quad (1)$$

where $|$ denotes the concatenation of two features, f is the *ReLU* activation function, $W_1 \in \mathbb{R}^{h\times 1024}$, $W_2 \in \mathbb{R}^{1024\times h}$ are the weights of the linear layers, b_1, b_2 are the bias terms, and h is the dimension of the hidden features. The fusing feature $X_F \in \mathbb{R}^{1024}$ is obtained by the element-wise product between the concatenated feature and X_{weight}. Finally, the weighted feature X_F is separated into new eyes feature $X_{eyes\_w} \in \mathbb{R}^{512}$ and new mouth feature $X_{mouth\_w} \in \mathbb{R}^{512}$:

$$X_F = [X_{eyes}|X_{mouth}] \otimes X_{weight} \quad (2)$$

$$X_{eyes\_w}, X_{mouth\_w} = X_F[:512], X_F[512:]. \quad (3)$$

Note that we exclude the *softmax* function here since a feature may thus dominate the others due to the exponential operation. Next, the weighted feature X_F can be used to predict a micro-expression label and to align the AU features.

3.5 AU Alignment

Action Units (AUs) are used to describe the movements of facial muscles. In other words, a model should extract similar facial features for samples having similar AU labels. This inspires us to design an auxiliary task to align the features with AU labels.

A simple method for incorporating AU labels into training is to use the same model but different heads for predicting AUs. However, directly predicting AUs is likely to face an imbalanced issue. To

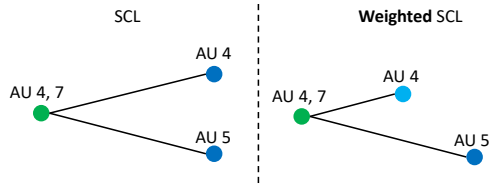

Figure 4: The comparison between SCL and weighted SCL (WSCL). Each sample is represented by a point, and the number above are the AU labels. Compared to SCL, WSCL handle the overlapping AU labels well and clusters the samples which have similar AU labels.

get rid of the imbalanced problem, we design a task based on AU similarity. Using a similarity score only requires comparing features and can avoid directly predicting the labels.

To train our model with similarity scores, we adopt Supervised Contrastive Learning (SCL) [11], which pulls and pushes the features at the class (AU) level. However, each sample can have multiple AU labels, and using SCL can only evaluate the similarity score by 0 and 1 and still use the AU labels as the ground truth. Therefore, using SCL cannot represent the relationship between the overlapping AU labels and still encounters the imbalanced problem. To solve the issues, we design Weighted SCL (WSCL) based on SCL to provide higher flexibility. To fit an similarity score into $[0, 1]$, we add an additional weight in $[0, 1]$ to describe the similarity score between two different samples. The ground truth of the similarity score w_{ij} between samples i and j is calculated as follows:

$$w_{ij} = \frac{|AU_i \cap AU_j|}{|AU_i \cup AU_j|}, \tag{4}$$

where AU_i and AU_j are the set of AUs for samples i and j, respectively. Using an similarity score as the ground truth can also prevent using the AU labels directly, which solves the imbalanced issue. To ensure that the training speed is not affected by computing w_{ij}, all the operations are vectorized. Figure 4 shows the comparison of two methods. In terms of a region-replaced sample, we mix up the AU labels from different samples to get the ground truth.

Finally, the loss function of WSCL is calculated as follows.

$$\mathcal{L}_{wscl} = -\sum_{i=1}^{N} \frac{1}{\sum_k w_{ik}} \sum_{j=1}^{N} \mathbf{1}_{\{i \neq j\}} w_{ij} \log \frac{exp(X_F^i X_F^j / \tau)}{\sum_{k=1}^{N} exp(X_F^i X_F^k / \tau)}, \tag{5}$$

where N is the batch size, X_F^i is the fusing feature in Eq. 2 of the i-th sample, and τ is the temperature parameter.

3.6 Classification

Before making a prediction, both X_{eyes_w} and X_{mouth_w} in Eq. 3 are sent into different double-layers fully-connected networks with a *softmax* function at the end to obtain X_{eyes_cls} and X_{mouth_cls}.

Moreover, since the class labels are also imbalanced, *i.e.*, the number of class *others* is nearly four times that of class *surprise* in CASME II and the number of class *anger* is almost five times greater than that of class *contempt* in SAMM, some research in facial-expression recognition adopts a weighted sampler to sample the same amount of data for different class labels within a batch. This method can be regarded as bootstrapping but with different sample probabilities for each data. Therefore, a weighted sampler might repeatedly sample out the same image. Due to the duplicated sampling, using a weighted sampler is not suitable for micro-expression recognition because the population of the training data is small, resulting in severely repeated sampling. Instead, to utilize all the training data properly, we balance the training loss with different class weights. A weight w_c for class c is computed by:

$$w_c = \max(n_1, n_2, \cdots, n_c)/n_c, \tag{6}$$

where n_k is the number of the training data of class k.

Finally, the classification loss is evaluated on the cross-entropy for both X_{eyes_cls} and X_{mouth_cls} by:

$$\mathcal{L}_{cls} = -\sum_{i}^{N} \sum_{c} w_c \mathbf{1}_{\{y_i=c\}} (\log(X^i_{eyes\_cls})_c + \log(X^i_{mouth\_cls})_c), \tag{7}$$

where N is the number of the samples, y_i is the ground truth of the ith sample, and $(X^i)_c$ is the cth value of the ith sample's output. The total loss function $\mathcal{L}_{total}$ is:

$$\mathcal{L}_{total} = (1 - \lambda)\mathcal{L}_{cls} + \lambda \mathcal{L}_{wscl}, \tag{8}$$

where λ is the hyperparameter controlling the importance of each training task. During the inference time, the ith predicted label is the argument that gives the maximum value of $X^i_{eyes\_cls} + X^i_{mouth\_cls}$.

4 EXPERIMENTS

4.1 Datasets

Following previous works, we evaluate the performance of the proposed model on five-class and three-class CASME II [38] and SAMM [3]. The details of each dataset are listed in Table 2.

CASME II. CASME II [38] collects 249 samples from 26 subjects at a frame rate of 200fps. Each image has a resolution of 640×480 pixels with around 280×340 pixels for the face area. There are five categories in CASME II—*disgust*, *happiness*, *repression*, *surprise*, and *others*. To have a fair comparison on three-class labels, we convert the data labels into *positive, negative*, and *surprise* by omitting the class *others*. The class *happiness* is changed into *positive*, and the *negative* class is made up of *disgust* and *repression*, and the *surprise* class remains unchanged.

SAMM. SAMM [16] collects 159 samples from 32 subjects at a frame rate of 200fps. The image resolution is 2040×1088 pixels, and the resolution of the face area is about 400×400 pixels. The five classes of SAMM are *contempt*, *happiness*, *anger*, *surprise*, and *others*. Likewise, for three-class comparison, we transform *happiness* into *positive* and transform *contempt, anger* into *negative*. The class *others* is removed, and the class *surprise* is not altered.

4.2 Evaluation Metrics

We employ Leave-One-Subject-Out (LOSO) cross-validation in all of our experiments. LOSO ensures that we choose one subject for testing in every training stage, and the rest of the subjects are used for the training. In order to compare with other works, we use

Table 2: Details of five-class and three-class labels in CASME II and SAMM.

Dataset	Five Categories					Three Categories		
CASME II	Disgust	Happiness	Repression	Surprise	Others	Positive	Negative	Surprise
	64	32	27	25	99	32	90	25
SAMM	Contempt	Happiness	Anger	Surprise	Others	Positive	Negative	Surprise
	12	26	57	15	26	26	92	15

Table 3: Comparison with other state-of-the-art methods on five-class and three-class CASME II and SAMM.

Models	CASME II			SAMM		
	UAR	Acc	UF1	UAR	Acc	UF1
Five Categories Comparison						
SSSN [10]	-	0.712	0.715	-	0.566	0.451
DSSN [10]	-	0.708	0.730	-	0.574	0.464
STRCN-A [35]	-	0.560	0.542	-	0.545	0.492
STRCN-G [35]	-	0.803	0.747	-	0.786	0.741
TSCNN-I [26]	-	0.741	0.733	-	0.635	0.607
TSCNN-II [26]	-	0.810	0.807	-	0.718	0.694
AU-ICGAN [36]	-	0.561	0.394	-	0.523	0.357
Graph-TCN [15]	-	0.740	0.725	-	0.750	0.699
FMER [14]	-	0.743	0.705	-	0.743	0.705
ours	0.850	**0.833**	**0.827**	0.751	**0.794**	**0.758**
Three Categories Comparison						
CapsuleNet [28]	0.702	-	0.707	0.599	-	0.621
NMER [17]	0.821	-	0.829	0.715	-	0.775
MTM [34]	-	0.756	0.701	-	0.741	0.736
AU-ICGAN [36]	-	0.712	0.355	-	0.702	0.433
STD [33]	-	0.799	0.759	-	0.767	0.764
FMER [14]	0.871	-	0.880	0.789	-	0.775
ours	**0.924**	**0.932**	**0.925**	**0.805**	**0.865**	**0.816**

accuracy and unweighted F1-Score (UF1) as the evaluation metrics:

$$Acc = \frac{\sum_i^S TP_i}{\sum_i^S TP_i + FP_i}, \tag{9}$$

$$UF1 = \frac{1}{C}\sum_i^C \frac{2 \times TP_i}{2 \times TP_i + FP_i + FN_i}, \tag{10}$$

where TP is the true positive, FP is the false positive, FN is the false negative, $\cdot_i$ is the metric for class i, C is the total number of classes, and S is the number of subjects. Additionally, as some other works use unweighted average recall (UAR) as the evaluation metrics, we also follow their settings to have a fair comparison:

$$UAR = \frac{1}{C}\sum_i^C \frac{TP_i}{n_i}, \tag{11}$$

where n_i is the number of samples of the class i.

4.3 Implementation Details

We implement the model using the PyTorch framework and train it on a Nvidia RTX 2080Ti GPU, having 12GB of memory. In terms of our model settings, we used AdamW Optimizer [21] with a learning rate of 3×10^{-3} and Cosine Annealing scheduler [20] with $T_0 = 10$ and $T_{mult} = 2$. For each LOSO training, we train our model for 100 epochs with a batch size of 128. We set the amplification factor to 6 and the hidden dimension h for AWM (mentioned in Section 3.4) to 1024 and λ to 0.3. To enhance the input diversity, we use data augmentation techniques such as `RandomHorizontalFlip`, `RandomErasing`, and `RegionReplacing` mentioned in Section 3.2. We also found that 1) using the *apex* frame instead of the amplified frame on CASME II and 2) using the amplified frame instead of the *apex* frame on SAMM to create the *On-A* and *Off-A* frames are better. One possible explanation is that CASME II has bigger differences between *onset–apex* frames and *offset–apex* frames than SAMM. Therefore, using an amplified frame to create optical flow leads to an obvious distortion on CASME II (average magnitude: 18 $\rightarrow$ 24), while increases the feature magnitude of SAMM (average magnitude: 8 $\rightarrow$ 14).

4.4 Experiments on CASME II and SAMM

We evaluate our model performance with other state-of-the-art methods on CASME II and SAMM. The results are shown in Table 3. As regards five-class comparison, our model improves the accuracy and F1-score of 2.83% and 2.47% on CASME II and 1.02% and 2.29% on SAMM. Due to less categories in three-class comparison, our improvements are apparent. We achieve 6.08% and 2.03% UAR improvement on CASME II and SAMM, respectively. The inferior performance of other methods may be caused by using whole facial areas as input. As the micro expressions are implicit, other models cannot decide which parts of the faces are more important. On top of that, other methods may easily be affected by imbalanced data. Our approaches can handle these imbalanced samples very well compared with other methods by Region Replacing and the help of AU Alignment.

4.5 Ablation Study

Here, we investigate the performance of our model components. "sb" represents the single-branch architecture (ResNet18 in our experiments) and "db" represents the dual-branch architecture proposed in our work. "AU" and "AWM" are AU alignment and Area Weighted Module. "all" indicates that both "AU" and "AWM" are used. Finally, "RR" denotes Region Replacing. The results of different experiments are shown in Table 4.

Region Replacing. We first remove RR from our training and examine the performances. The results indicate that the performances on both datasets drop, especially on SAMM. A possible reason is

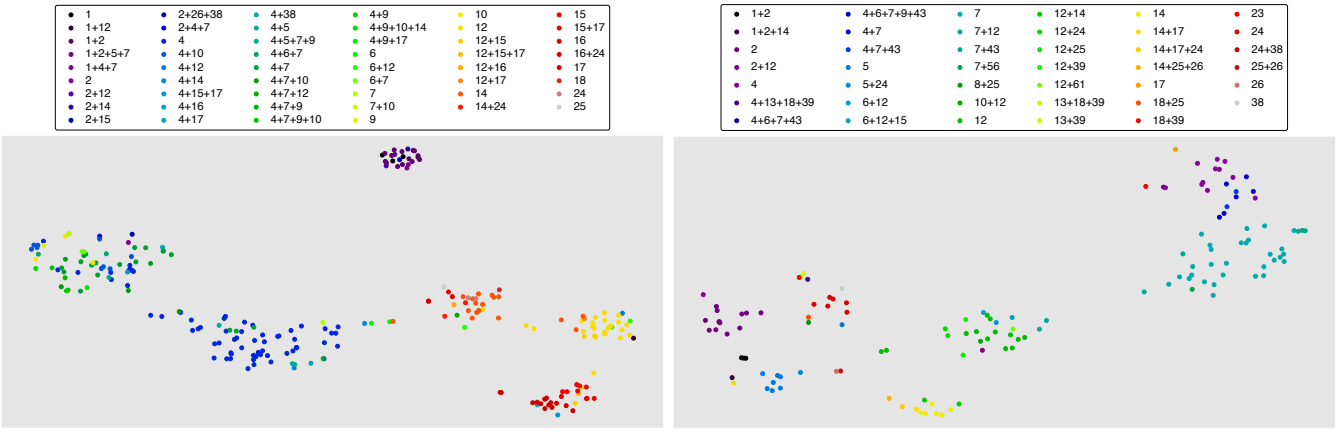

Figure 5: Distribution of AU features plotted on the (left) CASME II dataset and (right) SAMM dataset. Each point in the figure is a sample in the dataset. The color denotes the AU labels. We attach the legend on the top of the figures.

Table 4: Ablation study on model components. "sb" represents single-branch and "db" indicates dual-branch.

Models	CASME II		SAMM	
	Acc	UF1	Acc	UF1
sb	0.8008	0.7921	0.7574	0.7218
sb + AU	0.8008	0.7871	0.7206	0.6560
db	0.8008	0.7828	0.7500	0.6925
db + AU	0.8171	0.7974	0.7574	0.7338
db + AWM	0.8252	0.7957	0.7721	0.7347
db + all w/o RR	0.8171	0.8188	0.7647	0.7019
db + all	**0.8333**	**0.8267**	**0.7941**	**0.7582**

that SAMM has fewer overlapped AUs between the samples. Therefore, using RR provides more information to the SAMM dataset by changing the facial features and AU labels. This phenomenon is later shown and discussed in Section 4.6.1.

AU Alignment and AWM. Both AU alignment and AWM are designed to boost the performances. The former is to train our model with AU similarity scores to learn to capture the facial features, and the latter is to merge the area features with different weights. From the second part of Table 4, we observe that either AU alignment or AWM achieves better results than not using any of them. The final results reveal that using both components can further improve the performances. However, using AU alignment lowers the performances in the single-branch model. Our possible explanation is that single-branch architecture cannot extract variant patterns as dual-branch. Therefore, many facial patterns cannot be correctly extracted from the inputs; even some of the eye features may be extracted with the mouth patterns and vice versa, which may confuse the model[3].

[3]We concatenate the outputs from branch layers to form the AU features if AWM is removed and use the backbone outputs as the AU features in the single-branch model.

Dual-branch vs. Single-branch. Our proposed dual-branch architecture simulates the annotation process of the expressions and can also extract variant features from both eyes and mouth. From the upper part of Table 4, we first observe that single-branch approach outperforms the dual-branch approach on the SAMM dataset and has similar performance on the CASME II dataset without using AU alignment. This is expected because the dual-branch model should fuse the features with different weights, which different regions contribute unequally to the results. On the contrary, adding AU alignment to training makes the dual-branch model perform better than the single-branch model. We speculate that adding AU features provides the model with extra information to learn to extract essential features from the eyes and mouth, which mitigates the unweighted fusing problem.

4.6 Visualization

To better understand our model, we design two visualizations to observe 1) the efficacy of AU alignment and 2) how our model connects the AU features to the facial regions by distribution plot and AU-CAM, respectively.

4.6.1 Feature Distribution Plot. We include WSCL to align the AU features on similarity scores. Since the goal of WSCL is to pull the features of the samples with similar AU labels together and keep the features of different classes far away, we make distribution plots with weighted features X_F to show how AU features are aligned. The features are reduced to two-dimension with the t-SNE algorithm. Each point in Figure **??** represents a sample, and the color of each point represents different AU labels. The details of the labels are shown on the top of the distribution plot. For those samples having "+" signs in the legend, it has two or more AU labels. Also, we plot those similar AU labels with similar colors.

From the figure, we can observe that the points with the same color are clustered together, which shows that our model inherits the advantages from SCL. According to WSCL, a sample should be closed to similar class samples. For instance, in Figure **??**, the upper cluster, most in purple and black, includes the labels of AU1 and

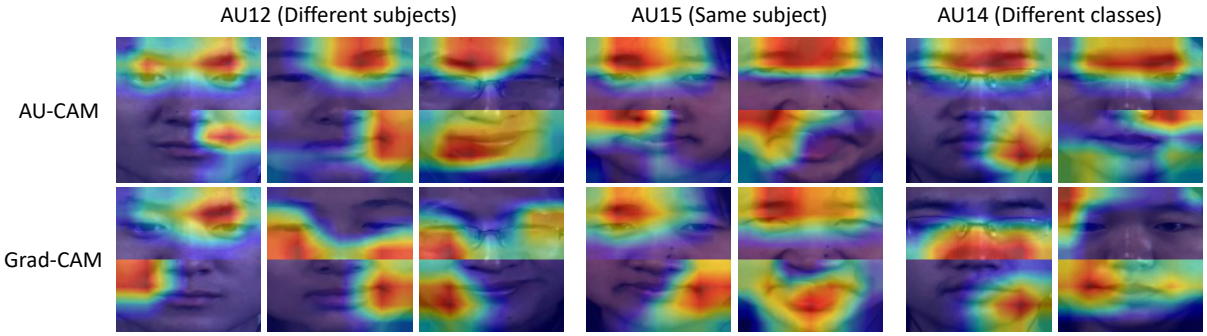

Figure 6: Comparison of AU-CAM and Grad-CAM in three scenarios. 1) Different subjects with the same AU label and emotion. 2) Different subjects with the same AU label and emotion. 3) Different subjects with the same AU label with different emotions.

AU2. Additionally, the points located at the half left part with the colors blue and green consist of samples having AU4 and AU7. On the other hand, in Figure **??**, samples in the colors of dark purple and black, placed at the left bottom, are made up of samples with AU1 and AU2, and most of the points in the upper right part, colored in dark blue and bright purple, have AU4 and AU7. Interestingly, although CASME II has more AU categories than SAMM, the AU labels usually overlap between the samples in CASME II. By contrast, each sample in SAMM has more inner-AU differences. This also explains why the points in Figure **??** have less overlapping between different categories than that in Figure **??**, and the performance differences with and without RR on SAMM is more prominent than that on CASME II (Section 4.5).

4.6.2 AU-CAM. CAM [42] provides a visualization method to which area interests the model most. Inspired by CAM, Grad-CAM [24] uses the gradients to determine the weights for each activation map. However, our goal is to **observe how the AU features are captured by our model** instead of knowing how our model decides the outputs. Therefore, we follow the way of finding the weights in Grad-CAM but with the different backward pass. Let C_{au-cam} be the results of AU-CAM:

$$C_{au-cam} = \text{VerticalConcat}(C_{eyes}, C_{mouth}), \tag{12}$$

$$C_b = \text{ReLU}(\sum_k \alpha_b^k A_b^k), \text{where } \alpha_b^k = \frac{\partial max(X_{b\_w})}{\partial A_b^k}, \tag{13}$$

where VerticalConcat is to vertically concatenate the eyes and the mouth images, b is "eyes" or "mouth", and A_b^k is the channel k output from the b branch. We select the maximum of $X._w$ in Eq. 3 to find out which AU feature contributes most to the facial regions.

Figure 6 shows the comparison between AU-CAM and Grad-CAM. We plot three different scenarios to discuss the results of our AU-CAM. Firstly, we select different subjects with the same AU label (AU12) and the same class (*happiness*). Our AU-CAM focuses on similar areas (eyes and mouth) among different subjects, while Grad-CAM shows little connection between the samples. Next, we provide the same-subject plot with the same emotion (*repression*) and the same AU (AU15). Likewise, AU-CAM captures similar areas in both eyes and mouth, but Grad-CAM shares only the eyes' information. Finally, we show the plots of the same-AU samples (AU14) with different subjects and emotions (*disgust* and *others*). Clearly, our AU-CAM highlights similar areas. In contrast, Grad-CAM provides less information regarding AU features and only shows how the expression labels are given.

We demonstrate how our AU-CAM captures the facial regions with AU features from these examples. In fact, AU-CAM is not limited to our works but can be extended to other applications. For instance, the AU features obtained from the graph structure can be used to backpropagate to obtain the AU-CAM. Besides, other facial tasks using AUs, such as macro-expression recognition or AU prediction, can also include AU-CAM to observe the interesting regions.

5 CONCLUSIONS AND FUTURE WORK

We propose a novel approach to recognizing the micro expressions by imitating the labeling strategy. We first introduce how we combine both amplified frame and optical-flow frames to form an input. Then, we present a dual-branch architecture for extracting the features from the eyes and mouth separately with Region Replacing. The region features are next combined with different weights computed by AWM. To utilize the AUs but avoid the imbalanced issue, we come up with Weighted SCL to align the AU features with the similarity scores. Experiments on the CASME II and SAMM datasets show that our model outperforms other state-of-the-art approaches, proving that MER can benefit from mimicking the annotation process. Also, to visualize the relationship between facial regions and AU features, we propose AU-CAM that can capture the activated regions caused by the AU features. In the future, we plan to incorporate graph structures into our dual-branch architecture to extract only the facial landmark features instead of the whole face for reducing the noise and boosting the inference speed.

ACKNOWLEDGMENTS

This work was supported in part by Ministry of Science and Technology of Taiwan under the grant numbers: MOST-109-2221-E-009-114-MY3, MOST-110-2221-E-A49-164, MOST-109-2223-E-009-002-MY3, MOST-110-2218-E-A49-018 and MOST-111-2634-F-007-002. We are grateful to the National Center for High-performance Computing for computer time and facilities.

REFERENCES

[1] Shaojie Bai, J Zico Kolter, and Vladlen Koltun. 2018. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv preprint arXiv:1803.01271* (2018).

[2] Aditya Chattopadhay, Anirban Sarkar, Prantik Howlader, and Vineeth N Balasubramanian. 2018. Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks. In *IEEE Winter Conference on Applications of Computer Vision*. 839–847.

[3] Adrian K Davison, Cliff Lansley, Nicholas Costen, Kevin Tan, and Moi Hoon Yap. 2016. Samm: A spontaneous micro-facial movement dataset. *IEEE Transactions on Affective Computing* 9, 1 (2016), 116–129.

[4] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2021. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *International Conference on Learning Representations*.

[5] Paul Ekman and Wallace V Friesen. 1978. Facial action coding system. *Environmental Psychology & Nonverbal Behavior* (1978).

[6] Vida Esmaeili, Mahmood Mohassel Feghhi, and Seyed Omid Shahdi. 2021. Micro-Expression Recognition Using Histogram of Image Gradient Orientation on Diagonal Planes. In *International Conference on Pattern Recognition and Image Analysis*. 1–5.

[7] Yiyang Gan, Ruize Han, Liqiang Yin, Wei Feng, and Song Wang. 2021. Self-supervised Multi-view Multi-Human Association and Tracking. In *ACM International Conference on Multimedia*. 282–290.

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 770–778.

[9] Miho Iwasaki and Yasuki Noguchi. 2016. Hiding true emotions: micro-expressions in eyes retrospectively concealed by mouth movements. *Scientific reports* 6, 1 (2016), 1–10.

[10] Huai-Qian Khor, John See, Sze-Teng Liong, Raphael CW Phan, and Weiyao Lin. 2019. Dual-stream shallow networks for facial micro-expression recognition. In *IEEE International Conference on Image Processing*. 36–40.

[11] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. 2020. Supervised contrastive learning. *In Neural Information Processing Systems* 33, 18661–18673.

[12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. ImageNet Classification with Deep Convolutional Neural Networks. *In Neural Information Processing Systems* 25.

[13] Ankith Jain Rakesh Kumar and Bir Bhanu. 2021. Micro-expression classification based on landmark relations with graph attention convolutional network. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 1511–1520.

[14] Ling Lei, Tong Chen, Shigang Li, and Jianfeng Li. 2021. Micro-Expression Recognition Based on Facial Graph Representation Learning and Facial Action Unit Fusion. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 1571–1580.

[15] Ling Lei, Jianfeng Li, Tong Chen, and Shigang Li. 2020. A novel graph-tcn with a graph structured representation for micro-expression recognition. In *ACM International Conference on Multimedia*. 2237–2245.

[16] Xiaobai Li, Tomas Pfister, Xiaohua Huang, Guoying Zhao, and Matti Pietikäinen. 2013. A spontaneous micro-expression database: Inducement, collection and baseline. *In IEEE International Conference on Automatic face and Gesture Recognition*, 1–6.

[17] Yuchi Liu, Heming Du, Liang Zheng, and Tom Gedeon. 2019. A neural micro-expression recognizer. In *IEEE International Conference on Automatic Face and Gesture Recognition*. 1–4.

[18] Yong-Jin Liu, Jin-Kai Zhang, Wen-Jing Yan, Su-Jing Wang, Guoying Zhao, and Xiaolan Fu. 2015. A main directional mean optical flow feature for spontaneous micro-expression recognition. *In IEEE Transactions on Affective Computing* 7, 4 (2015), 299–310.

[19] Ling Lo, Hong-Xia Xie, Hong-Han Shuai, and Wen-Huang Cheng. 2020. MER-GCN: Micro-expression recognition based on relation modeling with graph convolutional networks. In *IEEE Conference on Multimedia Information Processing and Retrieval*. 79–84.

[20] Ilya Loshchilov and Frank Hutter. 2017. SGDR: Stochastic gradient descent with warm restarts. In *International Conference on Learning Representations*.

[21] Ilya Loshchilov and Frank Hutter. 2019. Decoupled Weight Decay Regularization. In *International Conference on Learning Representations*.

[22] Tae-Hyun Oh, Ronnachai Jaroensri, Changil Kim, Mohamed Elgharib, Fr'edo Durand, William T Freeman, and Wojciech Matusik. 2018. Learning-based video motion magnification. In *European Conference on Computer Vision*. 633–648.

[23] Michel Owayjan, Ahmad Kashour, Nancy Al Haddad, Mohamad Fadel, and Ghinwa Al Souki. 2012. The design and development of a lie detection system using facial micro-expressions. In *International Conference on Advances in Computational Tools for Engineering Applications*. 33–38.

[24] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. 2017. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 618–626.

[25] Chidanand Shetty, Arooz Khan, Tanya Singh, and Keerti Kharatmol. 2021. Movie Review Prediction System by Real Time Analysis of Facial Expression. In *International Conference on Communication and Electronics Systems*. 1109–1113.

[26] Baolin Song, Ke Li, Yuan Zong, Jie Zhu, Wenming Zheng, Jingang Shi, and Li Zhao. 2019. Recognizing spontaneous micro-expression using a three-stream convolutional neural network. *IEEE Access* 7 (2019), 184537–184551.

[27] Aaron Van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *arXiv e-prints* (2018), arXiv–1807.

[28] Nguyen Van Quang, Jinhee Chun, and Takeshi Tokuyama. 2019. CapsuleNet for micro-expression recognition. In *IEEE International Conference on Automatic Face and Gesture Recognition*. 1–7.

[29] Haofan Wang, Zifan Wang, Mengnan Du, Fan Yang, Zijian Zhang, Sirui Ding, Piotr Mardziel, and Xia Hu. 2020. Score-CAM: Score-weighted visual explanations for convolutional neural networks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 24–25.

[30] Su-Jing Wang, Wen-Jing Yan, Xiaobai Li, Guoying Zhao, and Xiaolan Fu. 2014. Micro-expression recognition using dynamic textures on tensor independent color space. In *IEEE International Conference on Pattern Recognition*. 4678–4683.

[31] Yandan Wang, John See, Raphael C-W Phan, and Yee-Hui Oh. 2014. LBP with six intersection points: Reducing redundant information in LBP-TOP for micro-expression recognition. In *Asian Conference on Computer Vision*. 525–537.

[32] Hao-Yu Wu, Michael Rubinstein, Eugene Shih, John Guttag, Frédo Durand, and William T. Freeman. 2012. Eulerian Video Magnification for Revealing Subtle Changes in the World. *In ACM Transactions on Graphics* 31, 4 (2012), 1–8.

[33] Bin Xia and Shangfei Wang. 2021. Micro-Expression Recognition Enhanced by Macro-Expression from Spatial-Temporal Domain. In *International Joint Conference on Artificial Intelligence*. 1186–1193.

[34] Bin Xia, Weikang Wang, Shangfei Wang, and Enhong Chen. 2020. Learning from macro-expression: A micro-expression recognition framework. In *ACM International Conference on Multimedia*. 2936–2944.

[35] Zhaoqiang Xia, Xiaopeng Hong, Xingyu Gao, Xiaoyi Feng, and Guoying Zhao. 2019. Spatiotemporal recurrent convolutional networks for recognizing spontaneous micro-expressions. *In IEEE Transactions on Multimedia* 22, 3 (2019), 626–640.

[36] Hong-Xia Xie, Ling Lo, Hong-Han Shuai, and Wen-Huang Cheng. 2020. Au-assisted graph attention convolutional network for micro-expression recognition. In *ACM International Conference on Multimedia*. 2871–2880.

[37] Jingwei Yan, Jingjing Wang, Qiang Li, Chunmao Wang, and Shiliang Pu. 2021. Self-Supervised Regional and Temporal Auxiliary Tasks for Facial Action Unit Recognition. In *ACM International Conference on Multimedia*. 1038–1046.

[38] Wen-Jing Yan, Xiaobai Li, Su-Jing Wang, Guoying Zhao, Yong-Jin Liu, Yu-Hsin Chen, and Xiaolan Fu. 2014. CASME II: An improved spontaneous micro-expression database and the baseline evaluation. *In PloS ONE* 9, 1 (2014), e86041.

[39] Wen-Jing Yan, Qi Wu, Yu-Hsin Chen, Jing Liang, and Xiaolan Fu. 2013. How Fast Are the Leaked Facial Expressions: The Duration of Micro-Expressions. *In Journal of Nonverbal Behavior* 37, 4 (2013), 217–230.

[40] Qing-Long Zhang and Yu-Bin Yang. 2021. SA-Net: Shuffle attention for deep convolutional neural networks. In *IEEE International Conference on Acoustics, Speech and Signal Processing*. 2235–2239.

[41] Guoying Zhao and Matti Pietikainen. 2007. Dynamic texture recognition using local binary patterns with an application to facial expressions. *In IEEE Transactions on Pattern Analysis and Machine Intelligence* 29, 6 (2007), 915–928.

[42] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. 2016. Learning deep features for discriminative localization. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2921–2929.

[43] Hongyuan Zhu, Ye Niu, Di Fu, and Hao Wang. 2021. MusicBERT: A Self-supervised Learning of Music Representation. In *ACM International Conference on Multimedia*. 3955–3963.