Learning Interval-Aware Embedding for Macro- and Micro-expression Spotting

Xiaodong Li, Jiajun Li, Wenchao Du†, Hu Chen, and Hongyu Yang

College of Computer Science, Sichuan University, Chengdu, China
{lixiaodong, saranlee}@stu.scu.edu.cn,
{huchen, yanghongyu, wenchaodu.cs}@scu.edu.cn

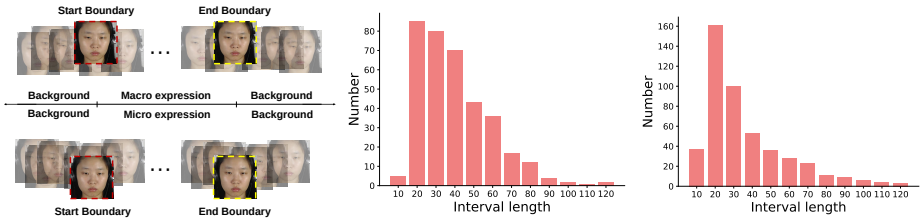
Abstract. Spotting the start and end frames of macro- and micro-expression in untrimmed long videos(i.e. Macro- and Micro-Expression Spotting, shorted by M^2ES) is extremely challenging due to the significant interval scale variations. Leading works borrowed the idea of "anchor" from temporal action localization into M^2ES, and achieved great improvements because of the finer proposal generation. However, covering diverse intervals is challenging for anchor-based methods due to latent domain shifts between macro- and micro-expression instances. Instead, we propose a purely anchor-free method for M^2ES, which eliminates the setting of redundant hyperparameters, and is both efficient and effective. In this work, we explore an Interval-aware Embedding Network (IAENet), which first exploits a basic two-stream network as the backbone to extract spatial and temporal feature embeddings from videos and optical flows, then a carefully designed temporal pyramid module is used to process interval-specific macro- and micro-expression instances in a parallel manner through a novel temporal attention mechanism and cross-scale feature fusion modules. We further design an interval-aware proposal generation scheme to specialize each spotting branch by sampling instances of proper intervals during training and inference. Extensive experiments demonstrate that our method beats all existing technologies, including interval-based and frame-based methods, with state-of-the-art results on the $CAS(ME)^2$ dataset and competitive results on the SAMM-LV dataset. Code is available.

Keywords: Macro- and Micro-Expression Spotting · Anchor-free · Temporal feature pyramid · Interval-aware proposal generation

1 Introduction

Facial expressions are an important form of non-verbal communication and can strongly reflect people's emotions. Depending on intensity and duration, they are generally divided into Macro-Expression (i.e. MaE, also known as regular or standard expression) and Micro-Expression (ME)[4]. MaE exhibits noticeable facial changes that typically span the entire face and last between 0.5 and 4

† Corresponding author.



(a) (b) (c)

Fig. 1: **(a)** The start and end moments of the macro- and micro expressions are indistinct, which cannot provide us with valuable information from frame-level features to spot the expression intervals. **(b)** Duration distribution statistics of the ground truth intervals (including MaE and ME) on the CAS(ME)^2 dataset. **(c)** Duration distribution statistics of the ground truth intervals (including MaE and ME) on the SAMM-LV dataset.

seconds. In contrast, ME involves subtle facial changes that often concentrate in specific facial regions and have a duration of less than 0.5 second [21]. Differing from MaE, which can be intentionally fabricated, ME is involuntary, spontaneously manifesting themselves[3]. In recent years, the extensive applications of ME in fields such as criminal investigations, clinical psychology, and business negotiations have attracted significant attention from the research community. Generally speaking, the facial expression analysis contains two implicit tasks: spotting and recognition, where the former is viewed as a prerequisite task for the latter. Therefore, Macro- and Micro-Expression Spotting (M^{2}ES), focusing on spotting the intervals of facial expressions from untrimmed long videos, is essential for the downstream facial expression recognition. Unfortunately, due to latent interval scale differences between MaE and ME, and inner characteristics of short duration and low intensity in ME, they pose great challenges in this field.

Deep-learning-based methods have made great progress on M^{2}ES in recent years, which can be primarily categorized into two classes: frame-based and interval-based. The former [15,33,11,26] typically involves generating probabilities for each frame and then designs a proposal generation strategy based on the probability maps to output proposals. Instead, interval-based methods[19,29,20,28] usually employ an effective encoding scheme to encode all sample intervals and directly generate proposals. These studies have shown promising performance on some benchmarks. However, frame-based methods always require scale-specific proposal generation strategies, which can be rigid and may not adapt well to various interval distributions. Interval-based methods depend on the complex modules specifically for encoding intervals, which lead to the necessity of frequent adjustment of hyperparameters.

Inspired by the ideas of "anchor-free" in the temporal action localization (TAL) field, we attempt to introduce the similar idea to alleviate the above problems. However, directly applying the existing frameworks to M^{2}ES is un-

reliable. As shown in Fig. 1, the inner interval scale variations between MaE and ME instances leads to significant sample imbalance distribution. Furthermore, the blurred spotting boundary for ME instances heavily limits the interval temporal representation learning.

To do so, in this work we propose a purely anchor-free M^{2}ES framework that directly learns Interval-Aware Embedding from untrimmed long videos (dubbed IAENet) to predict the possible offsets at each temporal point. The proposed IAENet makes dense predictions for the offsets along the temporal dimension in a parallel manner, which first extracts rich spatial and temporal features from the videos and corresponding optical flows to capture the facial appearance and motion information, then a carefully designed feature pyramid module is used to generate multiscale temporal embeddings. These embeddings would be fed into the head networks to predict the MaE and ME confidences and the interval-specific offsets of the start and end frames from each temporal location. To facilitate a more accurate boundary prediction, we carefully design an Interval-Aware Feature Pyramid module (termed by IAFP). Instead of aggregating the whole spatial facial expression variations, our IAFP aims to find the most salient moment-level feature for both start and end frames, i.e. start and end boundaries for each MaE and ME instances. As shown in Fig. 1a, the background regions near the start and end moments show indistinct boundaries, while spatial features inside the facial areas are almost the same, which cannot provide any information for judging if the MaE or ME starts or ends. This implies the importance of a moment-level feature.

Thus, we further inject the Temporal Attention Modules (TAM) and Cross-Scale Fusion Modules (CSFM) into the IAFP to enhance the temporal boundary representation ability of the embeddings. Moreover, as shown in Fig. 1b and Fig. 1c, the scale of ground truth intervals changes greatly on the two benchmark datasets, which makes it easier to generate proposals with extreme intervals during spotting. In order to address this issue, we also utilize an interval-aware proposal generation training scheme to make each branch specific to a given interval range matching its receptive field. Extensive experiments on the CAS(ME)^2[16] and SAMM-LV[25] datasets have demonstrated the effectiveness of our method, which improves by **2.67%** in terms of the F1-score compared to state-of-the-art methods on the CAS(ME)^2 dataset, and the results obtained on the SAMM-LV dataset are also comparable. In summary, the main contributions of our paper can be summarized as follows:

1) We present our investigation results on the effect of the interval variation in the M^{2}ES field. To our best knowledge, this is the first purely anchor-free M^{2}ES framework by exploring interval-aware embedding learning, which enjoys few hyperparameters tuning, more efficient sample interval generation, and better quantitative metrics compared to state-of-the-art methods.

2) A carefully designed temporal feature pyramid network is used to generate multiscale temporal embeddings, which couples the temporal attention module and cross-scale fusion module to process interval-specific macro- and micro-expression instances simultaneously.

3) Extensive experiments have demonstrated the effectiveness of our method, which achieves significant performance gains on the CAS(ME)^2 dataset, and competitive results on the SAMM-LV dataset.

2 Related Work

Traditional Spotting Methods. Conventional M^{2}ES assumes no large deformation of the face when no expression occurs, and aims to calculate differences within a fixed timescale, e.g., local binary patterns (LBP)[13], histogram of oriented gradient (HOG)[1], and optical flow [5]. However, the scales of expression vary enormously. For example, the longest interval on the CAS(ME)^2 dataset is more than 10 times longer against the shortest one. Therefore, a fixed temporal scale can only spot a few expressions. An intuitive approach is to employ multiscale durations, which in turn generate numerous negative samples due to the existence of general habitual movements, e.g., eye blinking, lip pursing and head shaking. This requires to cover more expressions, with fewer negative samples based on more scale variations.

Frame-based Spotting. Frame-based methods primarily generate interval proposals by predicting frame-level probabilities. SOFTNet[11] captured relevant features from different optical flow components to predict the likelihood of each frame belonging to a macro-expression or micro-expression. Leng et al. [7] utilized the MDMO [12] feature as input and evaluated the frame-level probabilities based on the input features. Yin et al. [27] encoded action units (AUs) label information into the network to enhance spatial feature embedding for obtaining probability sequences. Frame-based methods, after obtaining probability sequences, require the design of corresponding strategies to generate interval proposals. However, our purely anchor-free method can directly generate interval proposals by predicting the boundaries for each time position.

Interval-based Spotting. Interval-based methods directly generate interval proposals by encoding all sample intervals. Early methods commonly employed LSTM networks to encode temporal information. However, LSTM exhibits poor performance in handling complex and long time series. To address this issue, MESNet [21] introduced a specifically designed clip proposal module based on multi-level convolutional layers to encode intervals. LSSNet [28] and LGSNet[29] draw inspiration from the TAL [24] field and adopt a dual encoding approach, combining both anchor-based and anchor-free methods, to tackle the interval encoding problem. Unlike the above works that require the specialized design of encoding modules or frequent adjustment of numerous hyperparameters, our proposed purely anchor-free method encodes intervals in a simple and elegant manner, which is both efficient and effective.

Anchor-free Temporal Action Localization. The M^{2}ES and Temporal Action Localization (TAL) are similar to some degree, and the anchor-free methods have already garnered significant achievements in the TAL. AFSD [9], the first anchor-free model for temporal action localization, exploited boundary information to efficiently refine coarsely predicted proposals. MOC [8] achieved

efficient and precise localization performance by transforming action instances into the analysis and extension of trajectories for a motion point. BREM [6] introduced the boundary evaluation and region evaluation modules to obtain reliable proposals for improving the performance of TAL. However, directly applying these methods to M^{2}ES always leads to suboptimal results. The reason behind it is that the actions in the TAL task are mostly daily activities or sports activities, where action boundaries have distinct changes in background regions. Instead, the macro- and micro-expression boundaries do not present obvious changes, which makes the M^{2}ES task more challenging. To address this issue, we take inspiration from the above methods and propose an anchor-free framework tailored specifically for M^{2}ES.

Temporal Feature Pyramid Network. The initial concept of the feature pyramid network originated from the field of object detection. Most TAL methods primarily utilized downsampling to obtain temporal feature pyramid features. Actionformer [31] introduced several local window self-attention layers within the downsampling process to extract video representations. TriDet [17] further incorporated the SGP layer to mitigate issues associated with self-attention mechanisms, such as the ranking loss problem and high computational complexity. Although these methods utilized multiscale temporal information, they often ignored the effects of fusion modules among different scales. To address this limitation, we incorporate an additional upsampling operation to facilitate the more sufficient fusion of information between high-level and low-level temporal scale features.

3 Method

Given a video dataset denoted as $\mathcal{T} = \{\mathcal{T}_{train}, \mathcal{T}_{test}\}$, each data instance $\{X, \Psi\}$ contains a video $X=\{x_l\}_{l=1}^{L}$ with L frames. The corresponding annotation Ψ can be depicted by tuples $\{(\phi_m, y_m)\}_{m=1}^{M}$ where M is the number of MaE and ME instances in X, $\phi_m = (\psi_m, \xi_m)$ denotes the start frame and end frame, y_m indicates the MaE or ME category. Our goal is to train a model to predict intervals with class scores which could achieve high recall and precision with the ground truth on the test set $\mathcal{T}_{test}$.

Overview. The proposed anchor-free M^{2}ES framework(named IAENet) is shown in Fig. 2. Concretely, given a video X, we first process it to get its corresponding optical flows O. Then we exploit the pretrained feature extraction network to process the video, resulting in the corresponding video feature. The same operation is performed on optical flows O. Such features can contain the rich spatial and temporal information of the whole video, which are fused and then fed into the temporal feature pyramid module to generate multiscale embeddings. These embeddings are further utilized by the interval-specific prediction head modules to generate proposal sequences$\{(d\hat{s}_i, d\hat{e}_i, \hat{c}_i)\}$, where $d\hat{s}_i/d\hat{e}_i$ are offsets of the start and end frames and $\hat{c}_i$ means the confidence scores of the proposals.

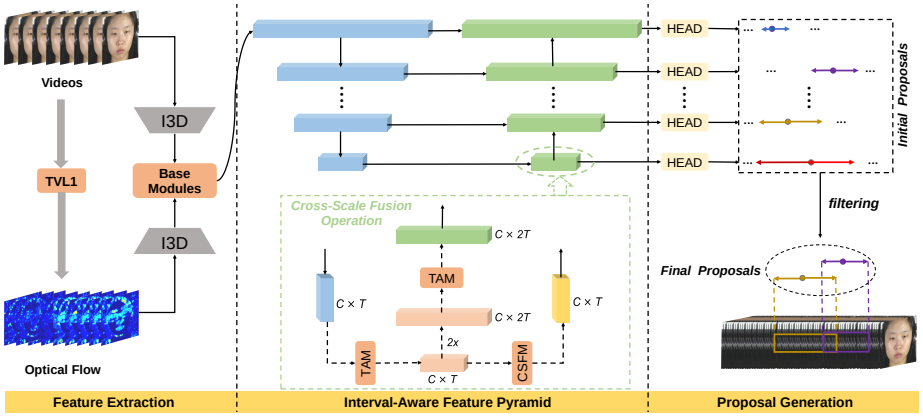


Fig. 2: Overview of IAENet. Firstly, we employ a pretrained I3D model to extract spatiotemporal feature embeddings from both the videos and the optical flow; These embeddings are then fed into the pyramid module to generate multiscale temporal embeddings; Finally, these multiscale embeddings are individually fed into the scale-corresponding prediction module to generate interval-specific proposals. TAM denotes the Temporal Attention Module, CSFM is the Cross-Scale Fusion Module, C is the dimension of the feature vector, T represents the temporal length of each layer.

3.1 Spatial-temporal Feature Extraction

To preserve facial motions without discarding appearance information, we extract spatial and temporal features from raw images and corresponding optical flow, where subtle and local movements in ME are important clues characterized by optical flow representation. Following [24], we first apply the TVL1 algorithm [22] to generate dense optical flows from input video X. Next, we use a pretrained I3D model [18] to extract feature maps $x_v \in \mathbb{R}^{N\times C}$ and $x_o \in \mathbb{R}^{N\times C}$ from X and O respectively, C denotes the dimension of the feature vector, and N is the number of segments. Assuming the length of the input video is denoted as L, the fixed length of the sliding window is w, and the number of overlapping frames among each sliding window is r, the number of segments can be calculated $N = \frac{L-w}{w-r} + 1$. In practice, w and r are set 8 and 6, respectively. Then, we concatenate x_v and x_o directly, followed by two stacked 1D convolutional layers and one max-pooling layer to achieve comprehensive fusion of information from both modalities.

3.2 Interval-Aware Feature Pyramid Module

Interval-Aware Feature Pyramid Module (IAFP) aims to acquire multi-scale temporal embedding, which serves for interval-aware embedding learning. As shown in Fig. 3, the proposed Interval-Aware Feature Pyramid Module primarily consists of three components: encoder, decoder, and cross-scale fusion operations. Encoder, composed of a stack of 1D convolutional layers with a $stride = 2$, is

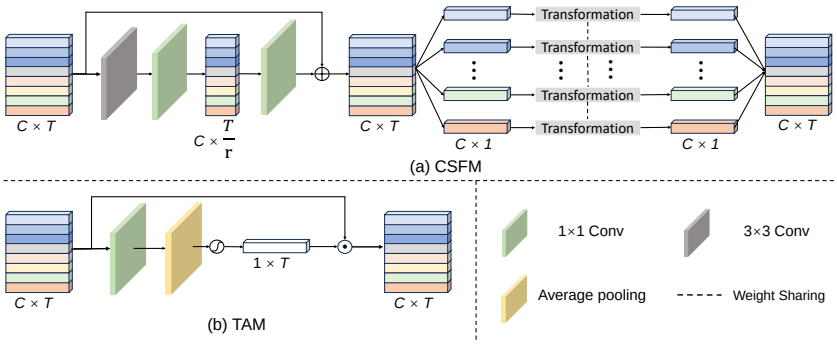


Fig. 3: (a) Cross-scale Fusion Module (CSFM). (b) Temporal Attention Module (TAM). C: the dimension of the feature vector. T: temporal length of each layer.

first used to compress the temporal features and generate multi-scale embeddings, where the embeddings from the deeper layers focus on the larger interval proposal prediction. Decoder aims to up-sample the encoded embeddings into original scales, which suppresses the global background information and leads to better interval boundary representation. In order to facilitate better interval embedding learning, we further propose novel cross-scale fusion operations, which include a Temporal Attention Module (TAM) and a Cross-Scale Fusion Module (CSFM).

Temporal Attention Module (TAM). TAM aims to enhance salient interval boundaries and facilitates the regression of genuine facial expression intervals. For simplicity, we denote the feature embeddings from the encoder as $F_i^e \in \mathbb{R}^{C\times T_i}$ and decoder embeddings as $F_{i+1}^d \in \mathbb{R}^{C\times T_{i+1}}$, where i represents the i-th layer, T_i denotes temporal length in the i-th layer, C is the dimension of the feature vector. As shown in Fig. 3(b), we utilize the simple but effective attention mechanism to implement TAM and employ it to enhance encoder and decoder feature embeddings,

$$F_i^d = Deconv(F_{i+1}^d) \tag{1}$$

$$F_i^{e'} = F_i^e \odot \sigma(Conv_{1\times1}(AP^{C\to1}(F_i^e))) \tag{2}$$

$$F_i^{d'} = F_i^d \odot \sigma(Conv_{1\times1}(AP^{C\to1}(F_i^d))) \tag{3}$$

where $DeConv(\cdot)$ denotes the deconvolution operation, $AP^{C\to1}$ is to involve an average pooling layer that compresses the channel dimension to 1, $Conv_{1\times1}$ represents a 1×1 convolution layer, and $\sigma(\cdot)$ is a *Sigmoid* function, and $\odot$ denotes the point-wise product operation.

Cross-Scale Fusion Module (CSFM). Considering directly fusing the temporal embeddings in our IAFP with a simple skip connection may lead to side effects due to latent semantic gaps, where the redundant background information could bring severe interference and blur the interval boundaries. Furthermore,

the cross-scale information is not exploited fully. To do so, we build a Cross-Scale Fusion Module to sequentially process information along the temporal and channel dimensions, thereby facilitating the information of interaction and fusion. As shown in Fig. 3(a), we first utilize the simple squeezing-and-excitation network to process the temporal dimension. Afterward, we exploit multiple linear layers, which share their weights, to perform the fusion of channel information at each temporal point.

3.3 Interval-aware Training and Inference

Label Assignment. Our interval-aware feature pyramid consists of several layers, each with a different regression range. We assign positive and negative samples based on the distance between the temporal point and the true boundary of the ground-truth interval and the regression range of each layer. Assuming that the ground-truth interval is defined as (S, E), the regression range of the i-th layer is (R_l^i , R_r^i), the j-th temporal point is denoted as T_j^i. First, we calculate the distances from the temporal point to the ground-truth interval boundaries, denoted as ΔS_j^i and ΔE_j^i. If both ΔS_j^i and ΔE_j^i are greater than 0, we select the larger value (assuming it is ΔS_j^i for demonstration purposes). If the larger value ΔS_j^i satisfies the following condition,

$$R_l^i \leq \Delta S_j^i < R_r^i \tag{4}$$

the temporal point T_j^i is labeled as a positive sample, which is further defined as $[\Delta S, \Delta E, C]$,

$$\Delta S = \Delta S_j^i / r \tag{5}$$

$$\Delta E = \Delta E_j^i / r \tag{6}$$

where r is the temporal stride of each layer, C denotes if it is MaE or ME. If any of the above conditions are not met, the temporal point is labeled as a negative sample.

Training. After going through the modules mentioned above, the initial spatial-temporal features generate the i-th layer feature F_i , which is then fed to the classification and regression heads for expression interval spotting. The total loss function is defined as:

$$L_{reg} = 1 - \frac{min(\hat{e}, e) - max(\hat{s}, s)}{max(\hat{e}, e) - min(\hat{s}, s)} \tag{7}$$

$$L_{all} = \frac{1}{N_{all}} \sum_{n=1}^{N_{all}} L_{cls} + \lambda \cdot \frac{1}{N_{pos}} \sum_{n=1}^{N_{pos}} L_{reg} \tag{8}$$

where L_{cls} utilizes a focal loss, $\hat{s}/\hat{e}$ represent the predicted start and end boundaries, s/e are the actual start and end boundaries, λ represents the ratio between the classification loss and the regression loss functions, N_{pos} and N_{all} denote the number of positive and all samples.

Inference. For each temporal point t, the regression head is responsible for regressing the distance to the left and right boundaries, outputting $[d\hat{s}, d\hat{e}]$. The classification head is responsible for classifying and outputting $[\hat{c}]$. Next, we follow the process below to obtain preliminary proposal results:

$$\hat{s} = t - d\hat{s} \cdot r \tag{9}$$

$$\hat{e} = t + d\hat{e} \cdot r \tag{10}$$

$$\hat{k} = \sigma(\hat{c}) \tag{11}$$

where $\hat{s}/\hat{e}$ means the start/end boundary of the proposal, r is the temporal stride of each layer, σ is a sigmoid function to ensure that the generated values are within the range of [0, 1]. The final proposal format is $[\hat{s}, \hat{e}, \hat{k}]$. Next, we use the top-k with k related to the maximum number of ground truth intervals to preliminarily filter out the top k proposals with the highest $\hat{k}$ values. Lastly, the NMS-based algorithm is employed to remove redundant proposals. Following previous work [28,29], we use the Non-Maximum Suppression (NMS) [14] on the CAS(ME)^2 dataset, and the Weighted Boxes Fusion (WBF) [2] on the SAMM-LV dataset.

4 Experiments

4.1 Datasets and Settings

Datasets. We conduct experiments on two benchmarks, i.e. CAS(ME)^2 and SAMM-LV. The CAS(ME)^2 dataset comprises 98 annotated videos from 22 subjects, with 30fps and 357 ground-truth instances, including 57 ME labels and 300 MaE labels. The original SAMM-LV dataset includes 224 long videos from 32 subjects, with 200fps and 499 ground-truth instances, including 159 ME labels and 340 MaE labels. We perform sampling on the SAMM-LV dataset to adjust its frame rate to 30fps, thus ensuring consistency in the frame rates across the two datasets. First, each video is divided into a series of non-overlapping segments with a length of 20 frames. These segments are further divided into three non-overlapping sub-segments, with lengths of 6, 7, 7 frames, respectively. Finally, only the first frame of each sub-segment is retained. In addition, we remove non-compliant intervals in the SAMM-LV dataset with a duration exceeding 4 seconds. Given that our sliding window size was set to 128 frames, we merge shorter videos of the same subject into a new video named "xx_99", where "xx" represents the subject's code number. Furthermore, for certain subjects, the total number of video frames does not reach the length of a sliding window. Hence, these videos are combined into a new subject named "99".

Evaluation Metrics. We employ Leave-One-Subject-Out (LOSO) cross-validation and a top threshold strategy in our experiments. A predicted proposal, i.e. W_{pr}, is considered a true positive sample when meeting the following conditions:

$$\frac{W_{pr} \bigcap W_{gt}}{W_{pr} \bigcup W_{gt}} \geq k_{IOU} \tag{12}$$

Table 1: F1-Score results on the CAS(ME)^2 and SAMM-LV datasets.

Methods	SAMM-LV			CAS(ME)^2		
	MaE	ME	Overall	MaE	ME	Overall
MDMD[5]	0.0629	0.0364	0.0445	0.1196	0.0082	0.0376
SP-FD[32]	0.0725	0.1331	0.0999	0.2131	0.0547	0.1403
He[30]	0.4149	0.2162	0.3638	0.3782	**0.1965**	0.3436
MESNet[21]	-	0.0880	-	-	0.0360	-
SOFTNet[11]	0.2169	0.1520	0.1881	0.2410	0.1173	0.2022
3D-CNN[26]	0.1595	0.0466	0.1084	0.2145	0.0714	0.1675
Concet-CNN[23]	0.3553	0.1155	0.2736	0.2505	0.0153	0.2019
MTSN[10]	0.3459	0.0878	0.2867	0.4104	0.0808	0.3620
ABPN[7]	0.3349	0.1689	0.2908	0.3357	0.1590	0.3117
AUW-GCN[27]	**0.4293**	0.1984	0.3728	0.4235	0.1538	0.3834
LGSNet[29]	-	-	**0.3880**	-	-	0.4360
Ours	0.3647	**0.2541**	0.3209	**0.5110**	0.1250	**0.4627**

where W_{gt} is the ground-truth, k_{IOU} is officially set to 0.5. According to the indicators of the MEGC2021 spotting track, we not only calculate the overall F1-Score, but also the F1-Score of ME and MaE.

Implementation Details. In the CAS(ME)^2 and SAMM-LV datasets, both RGB and optical flow are segmented with a duration of 8 frames and an overlap of 6 frames per segment. These segments are fed into the I3D model to extract features. During model training and evaluation, a sliding window of size 128 (i.e. the input features have a temporal dimension of 128) is utilized. We take Adam as the optimizer. The initial learning rate for CAS(ME)^2 is set to 0.0002, while for SAMM-LV it is set to 0.0004, and the cosine learning rate schedule is employed. The batch sizes for the CAS(ME)^2 and SAMM-LV datasets are both set to 128. The training epochs for both datasets are 30, including 5 warm-up epochs. All the experiments are performed on an NVIDIA RTX4090 GPU.

4.2 Comparision with State-of-the-art Methods

We compare IAENet with the representative methods on the CAS(ME)^2 and SAMM-LV datasets. These methods include traditional methods and deep-learning-based methods. Considering each method has exploited different post-processing strategies, it becomes challenging to achieve the same conditions for implementing these post-processing techniques. Therefore, we directly compare our results with those reported in their respective papers.

CAS(ME)^2. Tab. 1 lists the detailed results of different methods on the CAS(ME)^2 dataset. Our model exhibits clear superiority over other methods. Although our approach does not achieve the best results on the ME spotting, it still has made an 8.75% improvement for MaE. Nevertheless, our model achieves an overall F1-score improvement of 2.67% on the CAS(ME)^2 dataset, and achieving state-of-the-art results.

SAMM-LV. Tab. 1 also gives the quantitative results on the SAMM-LV dataset. Our method achieves comparable performances, but still falls short of

Table 2: Analysis of the number of pyramid layers on the two datasets.

Layers	SAMM-LV			CAS(ME)^2		
	MaE	ME	Overall	MaE	ME	Overall
#P1	0.3219	0.2251	0.2808	0.3874	0.1085	0.3520
#P3	**0.3417**	**0.2450**	**0.3090**	**0.4533**	0.1165	**0.4140**
#P4	0.3149	0.2282	0.2876	0.4368	**0.1379**	0.4012
#P5	0.2916	0.2390	0.2713	0.4307	0.1220	0.3927
#P6	0.2959	0.2230	0.2692	-	-	-

the state-of-the-art performance. Compared to the other methods, our model still exhibits unique advantages on ME spotting. However, the F1-score of MaE and overall F1-score fall behind. In Sec. 4.6, we analyze the possible causes behind it with the richer experiments and more detailed discussions.

4.3 Feature Hierarchy in Interval-Aware Feature Pyramid

Since we adopt an anchor-free method to encode all ground truth intervals, which means that we need to ensure to cover as many scales as possible. As shown in Fig. 1b and Fig. 1c, the duration of ground truth intervals on the CAS(ME)^2 dataset ranges from 8 to 117 frames, and the duration of ground truth intervals on the SAMM-LV dataset ranges from 5 to 118 frames, we fix the stride of the first pyramid layer to be 4 and 8, respectively. For convenient training, the regression range for each layer is set to $\left[\frac{S}{2}, 2S\right]$, where S is the stride of each layer. The minimum value of the regression range for the first layer is set to 0 and the maximum value of the regression range for the last layer is 128, to ensure that the scale of the generated proposals covers the scale of all ground truth intervals. The specific experimental data can be found in Tab. 2. Removing the feature pyramid (i.e. setting the number of pyramid layers to 1) resulted in a performance drop (-6.2% in overall F1-score on the CAS(ME)^2 dataset and -2.82% in overall F1-score on the SAMM-LV dataset). Next, we continuously increase the number of layers in the pyramid, starting from 3. The performance of our method generally increases with more pyramid levels, and is optimal when three layers are used on the two datasets.

4.4 Regression Range in Interval-aware Training

According to Sec. 4.3, there are three layers in our framework on the two datasets. To simplify, we only change the regression range of the first two levels. In detail, on the CAS(ME)^2 dataset, we set [0, R^{1st}_{cas}] in the 1st layer with R^{1st}_{cas} from 10 to 16. Likewise, the 2nd layer is set to [R^{2nd}_{cas}, 32] and R^{2nd}_{cas} ranges from 10 and 16. The regression range of the last levels is [16, 128]. On the SAMM-LV dataset, R^{1st}_{samm} in [0, R^{1st}_{samm}] is from 8 to 14, R^{2nd}_{samm} in [R^{2nd}_{samm}, 16] is from 4 to 8, the regression range of the last layer is [8, 128]. The detailed experimental results

Table 3: Different regression range of the first two layers on the two datasets.

Regression range	CAS(ME)^2 MaE	ME	Overall	Regression range	SAMM-LV MaE	ME	Overall
[[0,10], [10,32]]	0.4803	0.1319	0.4325	[[0,8], [4,16]]	0.3620	0.2329	0.3163
[[0,12], [10,32]]	0.4756	0.1061	0.4262	[[0,8], [6,16]]	0.3621	0.2389	0.3099
[[0,12], [12,32]]	0.4793	0.1000	0.4364	[[0,10], [4,16]]	0.3399	**0.2581**	0.2968
[[0,14], [10,32]]	0.4904	**0.1474**	0.4421	[[0,10], [6,16]]	0.3529	0.2467	0.3101
[[0,14], [12,32]]	0.4911	0.1148	0.4482	[[0,10], [8,16]]	0.3574	0.2385	0.3047
[[0,14], [14,32]]	**0.5110**	0.1250	**0.4627**	[[0,12], [4,16]]	**0.3647**	0.2541	**0.3209**
[[0,16], [12,32]]	0.4877	0.1207	0.4484	[[0,12], [6,16]]	0.3174	0.2508	0.2929
[[0,16], [14,32]]	0.4963	0.1290	0.4504	[[0,12], [8,16]]	0.3452	0.2500	0.3079
[[0,16], [16,32]]	0.4960	0.1026	0.4473	[[0,14], [8,16]]	0.3018	0.2500	0.2788

Table 4: Effects of the Interval-aware Feature Pyramid Module on CAS(ME)^2. Both denote using the Temporal Attention Module (TAM) and the Cross-Scale Fusion Module (CSFM).

Decoder	Both	MaE	ME	Recall	Precision	Overall
		0.4344	**0.1493**	0.3333	0.4760	0.3921
✓		0.4836	0.1111	0.3725	0.5320	0.4382
✓	✓	**0.5110**	0.1250	**0.3894**	**0.5697**	**0.4627**

can be found in Tab. 3. Considering the changes in the three indicators of MaE F1-score, ME F1-score and overall F1-score, the regression range of each layer on the CAS(ME)^2 dataset is [[0,14], [14,32], [16,128]], while the regression range of each layer in the SAMM-LV dataset is [[0,12], [4,16], [8,128]].

4.5 Ablation Study

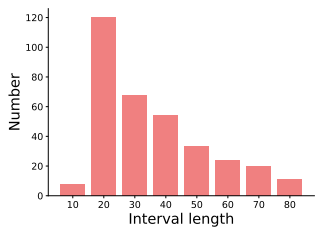
In this section, we conduct ablation experiments on the CAS(ME)^2 dataset to evaluate the impact of several key factors in IAENet. The number of pyramid layers and the regression range of each layer follow the optimal settings described in Sec. 4.3 and Sec. 4.4. Results are reported using pre-trained features with a fixed random seed for training.

Interval-aware Feature Pyramid. We use the encoder of the feature pyramid module as the baseline model. Then, we sequentially add the decoder and the cross-scale fusion operation into it. As shown in Tab. 4, introducing the decoder module leads to a significant overall performance improvement of 4.61%. This indicates that the fusion of features at different scales is benefit to improve the overall performance of the model. Next, we inject the cross-scale fusion operation into the decoder, which brings approximate 2.45% performance gains. Therefore, exploring interval-aware feature pyramid module is necessary to further promote the interaction and fusion of cross-scale spatial-temporal information.

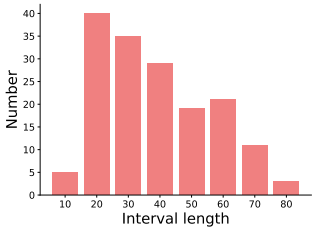
Interval-aware Training Scheme. We also evaluate the effect of the interval-aware training scheme and the results are shown in Tab. 5. After integrating the interval-scale training scheme, the model achieves significant improvements

Table 5: Analysis of the effectiveness of the interval-aware training scheme on CAS(ME)^2. "w/ IA" denotes our model uses the interval-aware training scheme. "w/o IA" means our model does not use the Interval-aware training scheme.

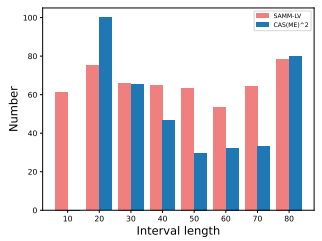
Name	MaE	ME	Recall	Precision	Overall	Layer1	Layer2	Layer3
w/ IA	**0.5110**	**0.1250**	**0.3894**	**0.5697**	**0.4627**	[7, 19]	[14, 53]	[20, 75]
w/o IA	0.4345	0.0816	0.3389	0.4708	0.3941	[5, 84]	[6, 106]	[6, 76]



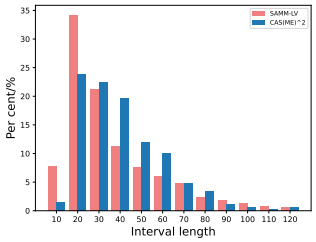
(a) FP samples



(b) TP samples



(c) Comparison of FP% on two datasets



(d) Interval distribution of two datasets

Fig. 4: Quantitative analysis on the SAMM-LV dataset

across various metrics. Additionally, we record the scale of the proposals generated at each layer under the two scenarios, as shown in the last three columns of Tab. 5. It can be observed that the inclusion of the interval-scale training scheme can effectively constrain the regression range of the proposals generated at each layer, which supports our model to learn interval-aware embedding better.

4.6 Quantitative analysis on SAMM-LV dataset

The proposed model achieves state-of-the-art (SOTA) results on the CAS(ME)^2 datasets. However, there is still a certain gap compared to the state-of-the-art results on the SAMM-LV dataset. In this section, we focus on analyzing the reasons for this gap specifically on the SAMM-LV dataset. We will visualize the data to provide a detailed examination of the observed differences and potential factors contributing to this gap.

In Fig. 4, the x-axis represents the length intervals with a unit of 10. For example, an x-axis value of 10 corresponds to the length interval $[0, 10]$. Firstly,

we visualize the distribution of false positive samples and true positive samples for proposals in Fig. 4a and Fig. 4b. It can be clearly observed that the distribution trends of the false positive samples and true positive samples are almost the same. More true positive samples correspond to more false positive samples. Additionally, we have also visualized the FP% metric for the CAS(ME)^2 and SAMM-LV datasets in Fig. 4c, respectively. Specifically, the FP% refers to the proportion of false positive samples among the total samples in each length interval unit. Except for a few length units, the FP% metric of the SAMM-LV dataset generally exceeds that of CAS(ME)^2. Furthermore, the FP% metric of each length unit in the SAMM-LV dataset generally exceeds 50%. The above two phenomena indicate that our model generates many false positive samples on the SAMM-LV dataset, which results in poor performance.

As shown in Fig. 4d, compared to the CAS(ME)^2 dataset, the distribution of the ground truth intervals in the SAMM-LV dataset is more concentrated in the shorter intervals. In our model, the shallower layers are responsible for the spotting of the shorter intervals. The shallower layers have a larger number of temporal points, which greatly increases the probability of generating false positive samples. At the same time, since the ratio of the medium-length intervals is relatively low, the model's spotting performance in these intervals is also poor. Finally, the subjects in the CAS(ME)^2 dataset are Asian, while the subjects in the SAMM-LV dataset are European. There are also subtle differences in facial expressions among different races, these factors may lead to our model does not perform well on the SAMM-LV dataset.

5 Conclusions

In this paper, we propose a purely anchor-free framework by learning interval-aware embedding to spot MaE and ME instances, termed by IAENet. To learn salient boundary features from untrimmed frames, we carefully design an interval-aware feature pyramid module, which exploits the temporal feature attention and cross-scale fusion modules to enhance interval-specific embedding learning. Moreover, we construct an interval-aware proposal generation scheme to constrain the proposal sample generation during training and inference for each spotting branch. Comprehensive experiments on two benchmark datasets demonstrate that our IAENet is effective and efficient, which achieves highly competitive performances with few hyperparameters tuning only.

Acknowledgements This work was supported by the National Natural Science Foundation of China (No. 62301345), the Natural Science Foundation of Sichuan Province of China (No. 2023NSFSC1403), the China Postdoctoral Science Foundation (No. 2023M732426), and the Fundamental Research Funds for the Central University of China (No. JCXK2233).

References

1. Davison, A., Merghani, W., Lansley, C., Ng, C.C., Yap, M.H.: Objective micro-facial movement detection using facs-based regions and baseline evaluation. In: 2018 13th IEEE international conference on automatic face & gesture recognition (FG 2018). pp. 642–649. IEEE (2018)
2. Du, D., Zhu, P., Wen, L., Bian, X., Lin, H., Hu, Q., Peng, T., Zheng, J., Wang, X., Zhang, Y., et al.: Visdrone-det2019: The vision meets drone object detection in image challenge results. In: Proceedings of the IEEE/CVF international conference on computer vision workshops. pp. 0–0 (2019)
3. Ekman, P.: Telling lies: Clues to deceit in the marketplace, politics, and marriage (revised edition). WW Norton & Company (2009)
4. Ekman, P., Friesen, W.V.: Nonverbal leakage and clues to deception. Psychiatry **32**(1), 88–106 (1969)
5. He, Y., Wang, S.J., Li, J., Yap, M.H.: Spotting macro-and micro-expression intervals in long video sequences. In: 2020 15th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2020). pp. 742–748. IEEE (2020)
6. Hu, J., Zhuang, L., Wang, B., Ge, T., Jiang, Y., Li, H., et al.: Estimation of reliable proposal quality for temporal action detection. arXiv preprint arXiv:2204.11695 (2022)
7. Leng, W., Zhao, S., Zhang, Y., Liu, S., Mao, X., Wang, H., Xu, T., Chen, E.: Abpn: Apex and boundary perception network for micro-and macro-expression spotting. In: Proceedings of the 30th ACM International Conference on Multimedia. pp. 7160–7164 (2022)
8. Li, Y., Wang, Z., Wang, L., Wu, G.: Actions as moving points. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVI 16. pp. 68–84. Springer (2020)
9. Lin, C., Xu, C., Luo, D., Wang, Y., Tai, Y., Wang, C., Li, J., Huang, F., Fu, Y.: Learning salient boundary feature for anchor-free temporal action localization. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 3320–3329 (2021)
10. Liong, G.B., Liong, S.T., See, J., Chan, C.S.: Mtsn: A multi-temporal stream network for spotting facial macro-and micro-expression with hard and soft pseudo-labels. In: Proceedings of the 2nd Workshop on Facial Micro-Expression: Advanced Techniques for Multi-Modal Facial Expression Analysis. pp. 3–10 (2022)
11. Liong, G.B., See, J., Wong, L.K.: Shallow optical flow three-stream cnn for macro-and micro-expression spotting from long videos. In: 2021 IEEE International Conference on Image Processing (ICIP). pp. 2643–2647. IEEE (2021)
12. Liu, Y.J., Zhang, J.K., Yan, W.J., Wang, S.J., Zhao, G., Fu, X.: A main directional mean optical flow feature for spontaneous micro-expression recognition. IEEE Transactions on Affective Computing **7**(4), 299–310 (2015)
13. Moilanen, A., Zhao, G., Pietikäinen, M.: Spotting rapid facial movements from videos using appearance-based feature difference analysis. In: 2014 22nd international conference on pattern recognition. pp. 1722–1727. IEEE (2014)
14. Neubeck, A., Van Gool, L.: Efficient non-maximum suppression. In: 18th international conference on pattern recognition (ICPR'06). vol. 3, pp. 850–855. IEEE (2006)
15. Pan, H., Xie, L., Wang, Z.: Local bilinear convolutional neural network for spotting macro-and micro-expression intervals in long video sequences. In: 2020 15th IEEE international conference on automatic face and gesture recognition (FG 2020). pp. 749–753. IEEE (2020)

16. Qu, F., Wang, S.J., Yan, W.J., Li, H., Wu, S., Fu, X.: Cas (me) ^{2}: a database for spontaneous macro-expression and micro-expression spotting and recognition. IEEE Transactions on Affective Computing **9**(4), 424–436 (2017)
17. Shi, D., Zhong, Y., Cao, Q., Ma, L., Li, J., Tao, D.: Tridet: Temporal action detection with relative boundary modeling. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 18857–18866 (2023)
18. Carreira, Joao and Zisserman, Andrew: Quo vadis, action recognition? a new model and the kinetics dataset. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 6299–6308. (2017)
19. Sun, B., Cao, S., He, J., Yu, L.: Two-stream attention-aware network for spontaneous micro-expression movement spotting. In: 2019 IEEE 10th International Conference on Software Engineering and Service Science (ICSESS). pp. 702–705. IEEE (2019)
20. Verburg, M., Menkovski, V.: Micro-expression detection in long videos using optical flow and recurrent neural networks. In: 2019 14th IEEE International conference on automatic face & gesture recognition (FG 2019). pp. 1–6. IEEE (2019)
21. Wang, S.J., He, Y., Li, J., Fu, X.: Mesnet: A convolutional neural network for spotting multi-scale micro-expression intervals in long videos. IEEE Transactions on Image Processing **30**, 3956–3969 (2021)
22. Wedel, A., Pock, T., Zach, C., Bischof, H., Cremers, D.: An improved algorithm for tv-l 1 optical flow. In: Statistical and Geometrical Approaches to Visual Motion Analysis: International Dagstuhl Seminar, Dagstuhl Castle, Germany, July 13-18, 2008. Revised Papers. pp. 23–45. Springer (2009)
23. Yang, B., Wu, J., Zhou, Z., Komiya, M., Kishimoto, K., Xu, J., Nonaka, K., Horiuchi, T., Komorita, S., Hattori, G., et al.: Facial action unit-based deep learning framework for spotting macro-and micro-expressions in long video sequences. In: Proceedings of the 29th ACM International Conference on Multimedia. pp. 4794–4798 (2021)
24. Yang, L., Peng, H., Zhang, D., Fu, J., Han, J.: Revisiting anchor mechanisms for temporal action localization. IEEE Transactions on Image Processing **29**, 8535–8548 (2020)
25. Yap, C.H., Kendrick, C., Yap, M.H.: Samm long videos: A spontaneous facial micro-and macro-expressions dataset. In: 2020 15th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2020). pp. 771–776. IEEE (2020)
26. Yap, C.H., Yap, M.H., Davison, A., Kendrick, C., Li, J., Wang, S.J., Cunningham, R.: 3d-cnn for facial micro-and macro-expression spotting on long video sequences using temporal oriented reference frame. In: Proceedings of the 30th ACM International Conference on Multimedia. pp. 7016–7020 (2022)
27. Yin, S., Wu, S., Xu, T., Liu, S., Zhao, S., Chen, E.: Au-aware graph convolutional network for macroand micro-expression spotting. In: 2023 IEEE International Conference on Multimedia and Expo (ICME). pp. 228–233. IEEE (2023)
28. Yu, W.W., Jiang, J., Li, Y.J.: Lssnet: A two-stream convolutional neural network for spotting macro-and micro-expression in long videos. In: Proceedings of the 29th ACM International Conference on Multimedia. pp. 4745–4749 (2021)
29. Yu, W.W., Jiang, J., Yang, K.F., Yan, H.M., Li, Y.J.: Lgsnet: A two-stream network for micro-and macro-expression spotting with background modeling. IEEE Transactions on Affective Computing (2023)
30. Yuhong, H.: Research on micro-expression spotting method based on optical flow features. In: Proceedings of the 29th ACM International Conference on Multimedia. pp. 4803–4807 (2021)

31. Zhang, C.L., Wu, J., Li, Y.: Actionformer: Localizing moments of actions with transformers. In: European Conference on Computer Vision. pp. 492–510. Springer (2022)
32. Zhang, L.W., Li, J., Wang, S.J., Duan, X.H., Yan, W.J., Xie, H.Y., Huang, S.C.: Spatio-temporal fusion for macro-and micro-expression spotting in long video sequences. In: 2020 15th IEEE international conference on automatic face and gesture recognition (FG 2020). pp. 734–741. IEEE (2020)
33. Zhang, Z., Chen, T., Meng, H., Liu, G., Fu, X.: Smeconvnet: A convolutional neural network for spotting spontaneous facial micro-expression from long videos. IEEE Access **6**, 71143–71151 (2018)