

Article

Micro-Expression Recognition Algorithm Using Regions of Interest and the Weighted ArcFace Loss

Peiying Zhang ¹ , Ruixin Wang ¹, Jia Luo ^{2,*} and Lei Shi ^{3,4,5}

¹ Qingdao Institute of Software, College of Computer Science and Technology, China University of Petroleum (East China), Qingdao 266580, China; zhangpeiying@upc.edu.cn (P.Z.); z24070100@s.upc.edu.cn (R.W.)

² College of Economics and Management, Beijing University of Technology, Beijing 100124, China

³ State Key Laboratory of Media Convergence and Communication, Communication University of China, Beijing 100024, China; leiky_shi@bupt.cn

⁴ State Key Laboratory of Intelligent Game, Yangtze River Delta Research Institute of NPU, Taicang 215400, China

⁵ Key Laboratory of Education Informatization for Nationalities (Yunnan Normal University), Ministry of Education, Kunming 650092, China

* Correspondence: jialuo@bjut.edu.cn

Abstract: Micro-expressions often reveal more genuine emotions but are challenging to recognize due to their brief duration and subtle amplitudes. To address these challenges, this paper introduces a micro-expression recognition method leveraging regions of interest (ROIs). Firstly, four specific ROIs are selected based on an analysis of the optical flow and relevant action units activated during micro-expressions. Secondly, effective feature extraction is achieved using the optical flow method. Thirdly, a block partition module is integrated into a convolutional neural network to reduce computational complexity, thereby enhancing model accuracy and generalization. The proposed model achieves notable performance, with accuracies of 93.96%, 86.15%, and 81.17% for three-class recognition on the CASME II, SAMM, and SMIC datasets, respectively. For five-class recognition, the model achieves accuracies of 81.63% on the CASME II dataset and 84.31% on the SMIC dataset. Experimental results validate the effectiveness of using ROIs in improving micro-expression recognition accuracy.



Academic Editor: Stefanos Kollias

Received: 23 November 2024

Revised: 18 December 2024

Accepted: 20 December 2024

Published: 24 December 2024

Citation: Zhang, P.; Wang, R.; Luo, J.; Shi, L. Micro-Expression Recognition Algorithm Using Regions of Interest and the Weighted ArcFace Loss.

Electronics **2025**, *14*, 2. <https://doi.org/10.3390/electronics14010002>

Copyright: © 2024 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article

distributed under the terms and

conditions of the Creative Commons

Attribution (CC BY) license

(<https://creativecommons.org/licenses/by/4.0/>).

Keywords: micro-expression recognition; convolutional neural networks; region of interest

1. Introduction

Micro-expressions (MEs) are rapid and subtle facial muscle movements that convey subconscious emotional responses. MEs are characteristically brief, with durations typically shorter than 1/5 of a second [1,2]. Despite their fleeting nature, they contain rich emotional information and are of great significance for understanding an individual's true emotional state. As a vital emotional recognition technology, ME recognition has extensive applications in fields such as psychology, medicine, and human–computer interaction [3,4]. Erroneous judgments may lead to misunderstandings; therefore, improving the accuracy and reliability of micro-expression recognition technology is of utmost importance.

In the current field of ME recognition, a multitude of challenges persist. First, the transient nature and subtlety of MEs make them difficult to detect, which often leads to the creation of datasets that are typically limited in size and lack diversity. Furthermore, differences in how individuals express the same emotion reduce the generalizability of

models. Additionally, the prevalent optical flow method is highly sensitive to lighting conditions and facial occlusions, which can severely degrade the performance of ME recognition. In applications such as public safety and mental health assessments, there is a high demand for ME recognition that prioritizes real-time performance. However, existing algorithms often struggle to achieve a balance between speed and precision.

In this paper, we use regions of interest to solve the problem of ME recognition. First, we analyze the difference between the onset frame and the apex frame to determine four ROIs as the input to the model. Then, we use the block partition module to process the input data based on the CNN. In addition, an improved ArcFace loss function is used to address the issue of data category imbalance in ME recognition. In summary, the contributions of this study are summarized as follows:

1. Based on optical flow information and ME action units, we select four ROIs and use the optical flow method for feature extraction. Through the study of local features in MEs, we identify that certain regions exhibit larger motion amplitudes during the occurrence of MEs, indicating richer ME information. Utilizing the ROIs for feature extraction results in more concentrated information, thereby reducing computational complexity and computation time. Additionally, to enhance feature extraction, this paper employs the TV-L1 method to capture deeper levels of feature information.
2. To enhance the efficiency of ME recognition, we use a block partition module. This module divides the optical flow graph and extracts its features for training the network. While ensuring recognition accuracy, the module also optimizes the network structure and reduces computational complexity, thereby improving the real-time performance of ME recognition. At the same time, we introduce a new WAF loss, which is based on ArcFace and processes angular features to deal with the imbalance of micro-expression datasets.
3. We perform experiments with three- and five-class schemes to assess the model's efficacy. Through ablation experiments, we demonstrate the effectiveness of the ROI method and the implemented module. Furthermore, our method shows improved performance compared to existing advanced methods, demonstrating its efficacy and potential for practical applications.

2. Related Works

In the 1960s, Ekman et al. [5] discovered the phenomenon of MEs through the analysis of facial expressions exhibited by a patient with severe depression. Building on this discovery, Ekman et al. [6] proposed the Facial Action Coding System (FACS). This system divides the face into multiple independent action units (AUs) based on the anatomical principles of the human face. Moreover, they defined six basic emotions (disgust, happiness, sadness, surprise, fear, and anger) through single AUs or their combinations. Through extensive research and analysis, Ekman et al. concluded that MEs are short-lived, subtle, and difficult to control. In order to improve the ability to recognize MEs, they developed the Micro-Expression Training Tool (METT) [7].

As research on ME recognition advances and manual identification becomes increasingly challenging, researchers have turned to computer vision techniques for assistance. Local binary patterns (LBPs) [8], a widely used feature extraction method in image processing and computer vision, have led to the development of several optical flow-based ME recognition methods, including STLBP-IP [9] and MLCNN-LBP [10].

Additionally, Shreve et al. [11] were among the first to derive optical flow and use its derivatives to extract optical flow features for ME analysis.

In recent years, deep learning has become increasingly prevalent in the field of ME recognition [12], garnering considerable research interest. Early on, Kim et al. [13] intro-

duced a CNN-LSTM architecture for ME recognition, utilizing a combination of tailored objective functions as constraints. Liong et al. [14] followed with a shallow triple-stream 3D CNN (STSTNet), leveraging optical flow attributes as input. Building upon these foundations, Zhang et al. [15] proposed a novel spatio-temporal transformer architecture that, to the best of our knowledge, was the first purely transformer-based approach for ME recognition. It eschewed the use of convolutional networks and achieved state-of-the-art performance by capturing both local and global spatio-temporal patterns. Similarly, Wang et al. [16] addressed the challenges of small facial expressions and dataset limitations by integrating micro-attention units into a ResNet architecture and employing transfer learning. Xia et al. [17] contributed to the understanding of MEs by noting that they only activate specific facial muscles related to local area changes. To address the issue of limited datasets, Peng et al. [18] suggested pre-training the network with larger datasets, complemented by data augmentation and oversampling techniques.

In more recent advancements, Li et al. [19] proposed the triple-branch attention fusion network for ME recognition (Triple-ATFME), which includes a preprocessing stage for optical flow feature extraction and a multi-path network with an adaptive channel attention module, outperforming conventional single-stream models. Wang et al. [20] introduced a Hierarchical Transformer Network (HTNet) that combines local self-attention mechanisms with an aggregation layer to effectively identify critical facial muscle movements in micro-expressions. Fan et al. [21] presented SelfME, a self-supervised learning framework that autonomously extracts facial motion, eliminating the need for traditional optical flow preprocessing, and developed SCViT to enhance feature learning by capturing symmetrical facial actions. Zhai et al. [22] introduced FRL-DGT, a framework that employs a self-supervised convolutional Displacement Generation Module for dynamic feature extraction and a transformer-based fusion mechanism to capture multi-level features for ME recognition.

3. Methods

In this section, we introduce the designed methodology. First, we select the regions of interest using dlib and then use optical flow to obtain features as input. Next, these features are processed through a block partition module and fed into the network for training. Finally, the WAF loss function is applied to the network as a method of data augmentation.

3.1. Feature Extraction

3.1.1. Regions of Interest

A region of interest [23] is a specific part of the image that needs to be focused on. In image processing tasks, selecting an ROI helps to extract and analyze features specifically from the designated region. By concentrating on ROIs, we can enhance processing efficiency and accuracy, leading to more effective image analysis.

AUs are a set of facial muscle movements defined in the FACS [24], which precisely depict the changes in facial expressions. The CASME II dataset [25] annotates the AUs corresponding to each micro-expression sample, where AUs 1, 2, 4, and 7 appear in the eyes and eyebrows; AUs 10, 12, 14, 15, 16, 25, and 26 appear around the mouth; AU 17 appears in the chin area; AU 6 appears in the cheeks; and AUs 9 and 38 appear in the nose area. For the distribution of AUs in different expressions, detailed data can be found in Table 1.

Through the analysis of AUs, we found that in many ME video sequences, there are several areas where movements are more pronounced: the eyes, eyebrows, and mouth. It can be inferred that these areas may contain more ME features.

In summary, we select four parts as the ROIs: the left eye combined with the left eyebrow, the right eye combined with the right eyebrow, the left corner of the mouth, and the right corner of the mouth.

Table 1. Distribution of AUs for different expressions.

Expression Label	AU
Happiness	1, 6, 12, 14, 15, 17
Disgust	2, 4, 7, 9, 10, 14, 15, 16, 17
Surprise	1, 2, 5, 7, 12, 14, 25, 26, 38
Repression	4, 12, 15, 17
Others	Corresponding regions for other expressions

In order to obtain the required four ROIs, we use the facial feature point detection technology provided by dlib. This technology can detect faces in images and mark 68 facial feature points. During the collection process of micro-expression datasets, due to variations in facial posture, angle, and irrelevant background, we need to perform face alignment.

Let the points (x_1, y_1) and (x_2, y_2) denote the positions of the inner canthi of the left and right eyes, respectively. The alignment procedure is determined by the following formula:

$$\begin{cases} x' = x_0 + (x - x_0) \cos \theta + (y - y_0)(-\sin \theta) \\ y' = y_0 + (x - x_0) \sin \theta + (y - y_0) \cos \theta \end{cases} \quad (1)$$

where $\theta = \arctan[(y_2 - y_1)/(x_2 - x_1)]$ is the angle of the face relative to the horizontal axis and (x', y') are the new coordinates post-rotation.

On this basis, we acquire the four ROI blocks required. We extracted 68 feature points of the face using dlib. Figure 1 shows an example face image after labeling 68 key feature points. These feature points include facial contour (0–16), left eyebrow (19–21), right eyebrow (22–26), nose (27–35), left eye (36–41), right eye (42–47), and mouth (48–67).

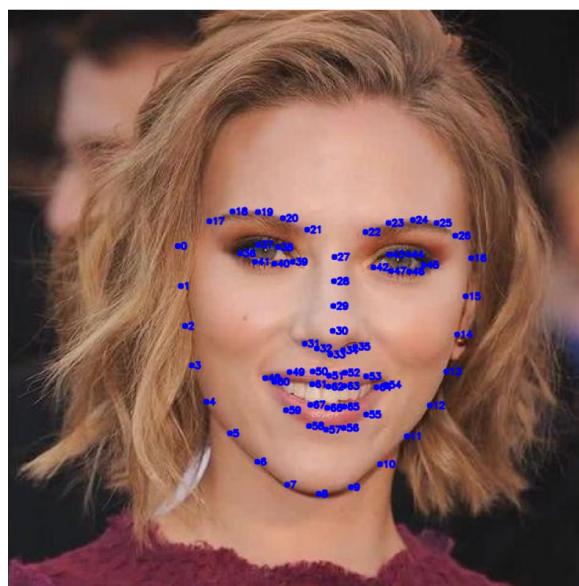


Figure 1. Example image showing the labeling of 68 key points.

Based on research on AUs and the location information of facial key points, we select the following key points as ROIs: left eye and left eyebrow (17–21, 36–41), right eye and right eyebrow (22–26, 42–47), left corner of the mouth (48–49, 59–60), and right corner of the mouth (53–55, 64).

We initially apply the optical flow method to the initial and peak frames to generate the optical flow images. Subsequently, the images are cropped based on the ROIs, resulting in four images, which serve as input data for model training. The relevant examples can be seen in Figure 2.

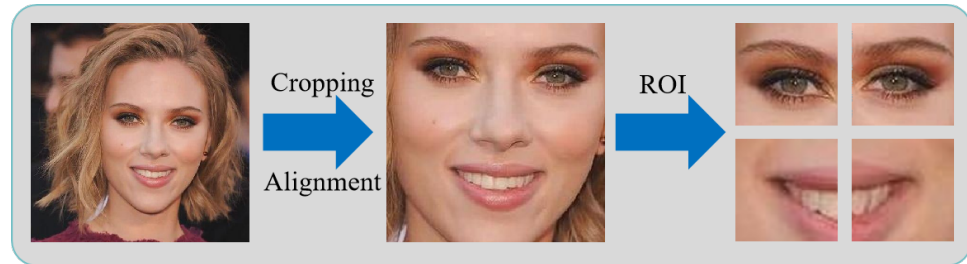


Figure 2. Regions of interest.

Before being sent to the block partition module, each cropped image is resized to a uniform size of 120×120 pixels. Then, these images are merged along the channel dimension to form a unified input dataset for training the network model.

3.1.2. Calculation of Optical Flow Characteristics

In the process of feature extraction, we aim to capture the motion features of MEs in videos. For this purpose, we pay special attention to the onset frame and apex frame of each video and use optical flow methods to achieve this goal. The optical flow method has been proven to make significant contributions in many ME recognition studies [26,27].

In this work, we use the TV-L1 optical flow method [28] to extract optical flow features. As is well known, optical strain [29] is commonly used to measure the deformation strength of optical flow vectors. In the context of optical flow techniques, u signifies the horizontal displacement within the flow field, v indicates the vertical displacement component, and os represents the optical strain (OS).

The basis of the optical flow method is the brightness-invariant equation, followed by operations such as differentiation. The brightness-invariant equation of the optical flow method is shown in Equations (2)–(4):

$$I_t(x, y, t) = I(x + dx, y + dy, t + dt), \quad (2)$$

$$(\nabla I)^T \vec{p} + I_t = 0, \quad (3)$$

$$\vec{p} = \left[p = \frac{dx}{dt}, q = \frac{dy}{dt} \right]^T, \quad (4)$$

where $I_t(x, y, t)$ represents the image intensity at point (x, y) at time t . dx, dy , and dt denote the displacement vectors of a point in the horizontal direction, vertical direction, and time, respectively. $\nabla I = (I_x, I_y)$ describes the spatial gradient of the image in the x-direction and y-direction. I_t is the temporal gradient of the image. p and q represent the motion vectors of a pixel point in the horizontal and vertical directions, respectively. The horizontal component u and the vertical component v of the optical flow field are determined using Equations (5) and (6):

$$p = \frac{dx}{dt} = \frac{\Delta x}{\Delta t} = \frac{u}{\Delta t}, u = p\Delta t, \quad (5)$$

$$q = \frac{dy}{dt} = \frac{\Delta y}{\Delta t} = \frac{v}{\Delta t}, v = q\Delta t. \quad (6)$$

Based on the obtained u and v , one can further extract the magnitude of the optical strain. The optical strain os is typically represented in the form of a matrix, as shown in Equation (7):

$$os = \begin{bmatrix} \epsilon_{xx} & \epsilon_{xy} \\ \epsilon_{yx} & \epsilon_{yy} \end{bmatrix} = \begin{bmatrix} \frac{\partial u}{\partial x} & \frac{1}{2} \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) \\ \frac{1}{2} \left(\frac{\partial v}{\partial x} + \frac{\partial u}{\partial y} \right) & \frac{\partial v}{\partial y} \end{bmatrix}, \quad (7)$$

where in the matrix, the diagonal terms ($\epsilon_{xx}, \epsilon_{yy}$) represent the normal strain components, while the off-diagonal terms ($\epsilon_{xy}, \epsilon_{yx}$) represent the shear strain components. Based on this matrix, the optical flow feature magnitude used is calculated using Equation (8):

$$|os| = \sqrt{\frac{\partial u^2}{\partial x} + \frac{\partial v^2}{\partial y} + \frac{1}{2} \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right)^2}. \quad (8)$$

We use the aforementioned optical flow extraction technique to capture the optical flow changes from the onset frame to the apex frame in ME sequences. Subsequently, this method is applied to transform the input initial and peak frames into four optical flow images corresponding to the regions of interest (ROIs), serving as the optical flow feature vectors for the network model input, denoted as $\theta = \{os_{leye_i}, os_{reye_i}, os_{lmouth_i}, os_{rmouth_i}\}$.

3.2. Network Architecture

Inspired by current research on ME recognition methods, we employ a block partition convolutional network. The network utilizes a block partition module that processes the input images by dividing them into blocks, calculating the feature information contained within each block, and training with the neural network based on this information. Consequently, it extracts key and effective information from the optical flow features obtained in the preceding sections.

3.2.1. Model Initialization and Parameter Configuration

The block partitioning convolutional network used can dynamically adjust its network structure according to the characteristics and scale of the input data. In the initialization stage of the model, we set the parameter size to control the number of output channels of the convolutional layer in the network, which affects the input feature dimension of the fully connected layer, allowing the model to adapt to input data of different resolutions and complexities.

In order to achieve this flexibility, we define a `net_dict` dictionary, which pre-sets the network parameters corresponding to different size values. By setting the size in training, the effect of changing other related parameters can be achieved. This design enables the block partition convolutional network model to dynamically adjust the number of channels of the convolutional layer (`conv2_0`) and the number of input features of the fully connected layer (`fc1_0`) according to actual application requirements and usage needs, and select the most appropriate network scale to balance computational efficiency and model performance.

3.2.2. Network Model Structure

The network contains two convolutional layers, one pooling layer, one view layer, and three fully connected layers. Its structure is shown in Table 2.

In the first convolutional layer (`conv1`), the number of input channels matches `feature_num`, and the number of output channels is fixed at 8. This layer is responsible for extracting preliminary low-level features from the input data.

Table 2. Network model structure.

Layer Name	Type	Input Size	Output Size
input	-	$N \times 3 \times H \times W$	-
conv1	Conv2d	$N \times 3 \times H \times W$	$N \times 8 \times H \times W$
pooling	MaxPool2d	$N \times 8 \times H \times W$	$N \times 8 \times H/3 \times W/3$
conv2	Conv2d	$N \times 8 \times H/3 \times W/3$	$N \times C_2 \times H/3 \times W/3$
view	View	$N \times C_2 \times H/3 \times W/3$	$N \times C_2 \times H/3 \times W/3$
fc1	Linear	$N \times C_2 \times H/3 \times W/3$	$N \times 128$
fc2	Linear	$N \times 1024$	$N \times 128$
fc3	Linear	$N \times 128$	$N \times class_num$

In the pooling layer (pool), maximum pooling is used, the pooling window size is 3, the step size is 3, and no padding is used. While performing dimensionality reduction operations, this layer increases the invariance to small translation changes.

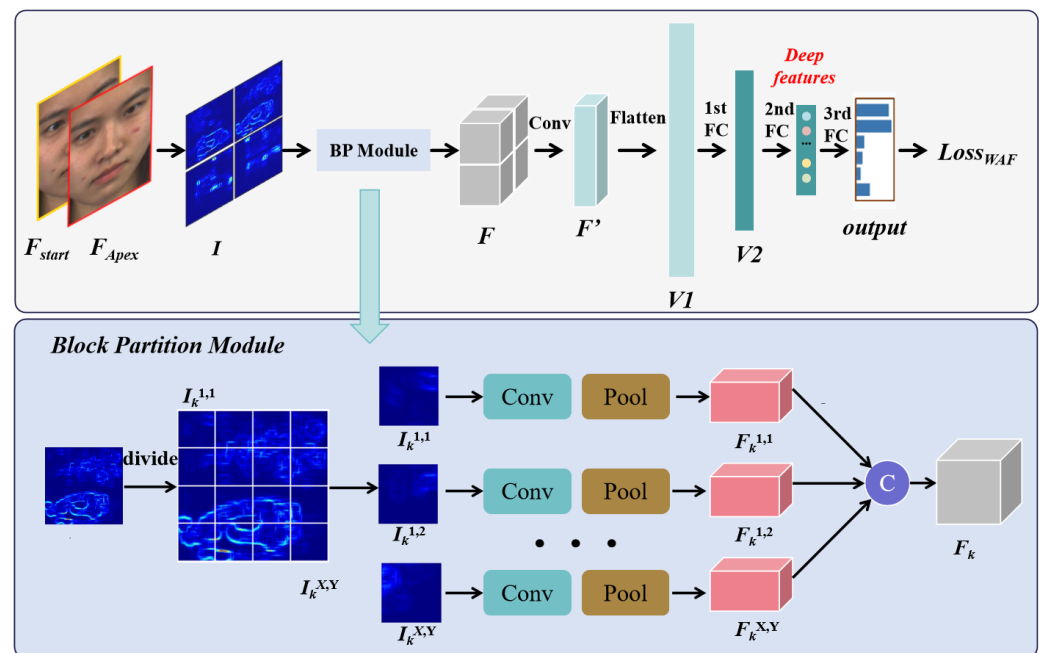
In the second convolutional layer (conv2), the number of input channels is determined by the output of the conv2_0 layer, and the number of output channels is determined by conv2_1, which further refines and abstracts the feature representation.

The view layer (view) adjusts the shape of the input tensor in the previous layer to ensure that the input shape of the subsequent fully connected layer is correct.

In the fully connected layer (fc), three fully connected layers (fc1, fc2, and fc3) are connected in sequence. After flattening, the input is sequentially reduced through these three fully connected layers, and finally, the required output is obtained.

3.2.3. Network Execution Process

As previously mentioned, the input to the block partition convolutional network is the four optical flow feature images extracted from the aforementioned ROIs, and the corresponding image features are obtained from them through the corresponding feature extraction path. Figure 3 shows a flowchart of the network.

**Figure 3.** Network flowchart and block partition module

After obtaining the input image, it must first pass through the block partition module shown in the figure to calculate the features of each image. Let $I_k \in R^{H \times W \times C}$ represent

the image of the divided ROI area, where H represents the image height, W represents the image width, and C represents the number of channels. I_1, I_2, I_3 , and I_4 are the optical flow features of the four images after the region of interest is divided.

In the execution of this module, the input is initially divided into tiles measuring $S \times S$ pixels, resulting in $X \times Y$ tiles, where $X = \frac{W}{S}$ and $Y = \frac{H}{S}$. The partitioning process is detailed in Equation (9):

$$I_k = \{I_{k^n}^{i,j} \mid i \in [1, X], j \in [1, Y]\}, \quad (9)$$

where $I_{k^n}^{i,j}$ denotes a block situated at row i and column j within the feature map I_k , with dimensions $S \times S \times C$.

For feature extraction, each block is processed by an 8-kernel 3×3 convolutional layer, followed by a 3×3 max pooling layer to reduce the spatial dimensions and select key features. The calculation process is shown in Equations (10) and (11):

$$L_k^{i,j} = conv_0(I_k^{i,j}), \quad (10)$$

$$F_k^{i,j} = maxpool(L_k^{i,j}), \quad (11)$$

where $conv_0(.)$ represents the 1×1 convolution operation. Each block $I_k^{i,j}$ is convolved to obtain the output $L_k^{i,j}$. After maximum pooling, it obtains $F_k^{i,j}$, where the size of $F_k^{i,j}$ is $S' \times S' \times C'$.

Afterward, according to its channel dimension, $F_k^{i,j}$ performs a concatenation operation, as shown in Equation (12):

$$F_k = concat(F_k^{11}, F_k^{12}, \dots, F_k^{X,Y}). \quad (12)$$

In this framework, the aggregated feature F_k for the k -th image is constructed by integrating the features from individual partitions, resulting in a comprehensive feature representation. The dimensions of F_k are $S' \times S' \times (C' \times X \times Y)$, which encapsulates the combined information from all partitions. Following the partitioning of each image through the block partition module, the four distinct partition features are consolidated into an encompassing feature set F , characterized by the dimensions $F \in \mathbb{R}^{S' \times S' \times (4C' \times X \times Y)}$, as shown in Equation (13):

$$F = concat(F_1, F_2, F_3, F_4). \quad (13)$$

F summarizes the optical flow feature vectors contained in each block. In the ME recognition task, not all features contained in all blocks have the same importance. Therefore, the obtained feature set F needs to be screened to retain useful information. First, we reduce the dimensions of F using a 1×1 convolution operation. The calculation method is shown in Equation (14):

$$F' = conv_1(F), \quad (14)$$

where $conv_1(.)$ represents the 1×1 convolution operation used to adjust the number of feature map channels. After this operation, the output is $F' \in \mathbb{R}^{S' \times S' \times C_1}$, where C_1 represents the number of channels after convolution dimensionality reduction. Then, in order to convert the multi-dimensional feature map into a one-dimensional vector that is more suitable for processing by the fully connected layer, a flattening operation is performed to obtain a one-dimensional vector V_1 with a length of $1_1 = S' \times S' \times C_1$. Finally, V_1 is further processed by two layers of fully connected layers to obtain its deep features. The calculation method is shown in Equations (15) and (16):

$$V_2 = \sigma(W_1 \times V_1), \quad (15)$$

$$a = \sigma(W_2 \times V_2), \quad (16)$$

where the σ is the ReLU activation function, and W_1 and W_2 are the weights of the corresponding fully connected layer. V_2 is the output of the first fully connected layer, with a length of l_2 . a is a deep feature, which represents the outcome from the subsequent fully connected layer, characterized by a length of l .

Throughout the training phase, the dataset $T = \{(\theta_i, y_i)\}$ employs the network G with parameters w for the learning procedure, where y_i denotes the label associated with the i -th instance of features among E categories. Utilizing the aforementioned method, we derive the output vector $\mathbf{o}_i = [o_{i1}, \dots, o_{iE}]^T = G(\theta_i, w)$. Subsequently, the parameters are adjusted based on the calculated loss.

After obtaining the corresponding deep feature $\mathbf{o}$, it passes through the third fully connected layer. The weight of this connection layer can be calculated by the loss function described below. The output result after passing through this connection layer is an E -dimensional vector *output*, where E is the number of categories for ME recognition, which is 3 or 5 in this article. Finally, the predicted label result can be obtained based on the *output*.

3.3. WAF Loss

Deng et al. [30] proposed ArcFace, a loss function that yields highly discriminative features. This approach leverages the cosine similarity to align features from the same class while maintaining a distinct gap with features from other classes. By expanding the angular margin within the cosine metric, ArcFace enables the model to more precisely distinguish between category boundaries, thereby boosting the precision and reliability of the classification process. The calculation method is shown in Equation (17):

$$L = -\frac{1}{N} \sum_{i=1}^N \log \frac{e^{s(\cos(\theta_{y_i} + m))}}{e^{s(\cos(\theta_{y_i} + m))} + \sum_{j=1, j \neq y_i}^n e^{s \cos \theta_j}}, \quad (17)$$

where N denotes the aggregate sample count within the training dataset, s represents a scaling parameter that regulates the magnitude of the gradient in the loss function, and m refers to the cosine margin, employed to introduce extra spacing within the angular domain. y_i signifies the class-specific covariance matrix, which is derived from the class y_i .

Based on the aforementioned content, we implement a new loss function, referred to as the Weighted ArcFace (WAF) loss. This function, based on the ArcFace loss function, includes weights for E different micro-expression categories, which can be adjusted according to the needs of specific tasks.

With the addition of these weights, the function not only serves to enhance deep features but also effectively addresses the disparity in class representation inherent to ME datasets. The methodology for calculating this loss function is shown in Equations (18) and (19):

$$L_j = -\frac{1}{N} \sum_{i=1}^N \log \frac{e^{s(\cos(\theta_{y_i} + m))}}{e^{s(\cos(\theta_{y_i} + m))} + \sum_{j=1, j \neq y_i}^n e^{s \cos \theta_j}}, \quad (18)$$

$$L = \sum_{j=1}^E \mu_j L_j, \quad (19)$$

where N signifies the count of instances pertaining to class j and μ_j denotes the loss weight value for category j .

The architectural framework along with the WAF loss function is outlined in Algorithm 1. The dataset D , defined as $\{(f_{oi}, f_{ai})\}, 1 \leq i \leq N$, represents the initial collection of data, with f_{oi} and f_{ai} indicating the onset and apex frames for the i -th sample, respectively.

Algorithm 1 WAF loss.

Input: D : initial dataset; μ : loss weight;
Output: W : fc3 weight; b : fc3 bias; w : network weight;
1: **for all** onset frames and Apex frames $f_{oi}, f_{ai} \in D$ **do**
2: Calculate os_i using Equations (2) to (8);
3: $\theta = \{os_leye, os_reye, os_lmouth, os_rmouth\}$;
4: Add θ_i to T ;
5: **end for**
6: Construct $train_{set} T = \{(\theta_i, y_i)\}$.
7: Randomly assign values to W, b , and w .
8: **for** epoch from 1 to max_epoch **do**
9: Generate $o_i = G(\theta_i, w)$.
10: Determine L_j via Equation (18).
11: Aggregate the total loss L as per Equation (19).
12: Update the parameters W, b , and w .
13: **end for**
14: **return** W, b , and w .

4. Experiments

Experiments were carried out utilizing three distinct datasets: the SMIC [31], SAMM [32], and CASME II [25] datasets. These datasets are extensively utilized in the field of micro-expression recognition tasks.

CASME II: This dataset originates from the Institute of Psychology at the Chinese Academy of Sciences and includes micro-expression instances from 24 Asian subjects. It contains 256 micro-expression instances. The data were captured at a rate of 200 frames per second, with a resolution of 640×480 pixels and a facial area size of approximately 280×340 pixels. The CASME II dataset includes seven types of micro-expressions: “happiness”, “surprise”, “disgust”, “sadness”, “fear”, “repression”, and “others”.

SMIC: This dataset is compiled from a simulated interrogation room environment and includes micro-expression samples from 16 participants from Europe, Asia, and Africa. It contains 164 micro-expression instances. The data were captured at a rate of 100 frames per second, with a resolution of 640×640 pixels and a facial area resolution of approximately 190×230 pixels. The SMIC dataset encompasses three types of micro-expressions: “negative”, “positive”, and “surprise”.

SAMM: This dataset originates from a highly controlled laboratory setting and comprises micro-expression instances from 28 subjects across 14 different ethnicities, including those from Europe, Asia, and Africa. It encompasses 159 instances of micro-expressions. The data were recorded at a frame rate of 200 frames per second, with an image resolution of 2040×1088 pixels and a facial area dimension of approximately 400×400 pixels. The SAMM dataset includes eight distinct types of micro-expressions: “happiness”, “surprise”, “disgust”, “fear”, “contempt”, “angry”, “repression”, and “others”.

The three datasets utilized included as diverse a range of data from different regions as possible, encompassing a total of 68 subjects (CASME II: 26; SAMM: 28; SMIC: 16). To standardize the emotion classification across these varied datasets, we aligned our approach with the MEGC2019 Challenge [33] criteria for data categorization. For the three-class task, we categorized the emotions based on the labels provided in the original datasets as follows: positive includes “happiness”; surprise includes “surprise”; and negative includes “fear”,

“anger”, “disgust”, “sadness”, “contempt”, and “repression”. For the five-class task, the emotions were categorized as happiness, surprise, disgust, repression, and others.

4.1. Experimental Results

We used the leave-one-subject-out (LOSO) cross-validation method to assess performance. LOSO cross-validation involves selecting the samples of one subject as the test set and the samples of the remaining subjects as the training set in each round of experiments, repeating this process until each subject’s samples have been used as the test set once. Given that we had a total of 68 subjects across the three datasets, it was necessary to conduct 68 rounds of experiments.

Owing to the uneven distribution of sample sizes across various emotional categories within the dataset, we adopted accuracy (ACC), unweighted F1 score (UF1), and unweighted average recall (UAR) as evaluation metrics. The mathematical expressions for these metrics are as follows:

$$Accuracy = \frac{1}{k} \sum_{i=1}^k Accuracy_i, \quad (20)$$

$$UF1 = \frac{1}{C} \sum_{i=1}^C \frac{2 \times TP}{2 \times TP + FP + FN}, \quad (21)$$

$$UAR = \frac{1}{C} \sum_{i=1}^C \frac{TP}{TP + FN}, \quad (22)$$

In this experiment, k denotes the number of participants, with $k = 68$. The parameter C (where C can be 3 or 5 in this context) signifies the number of distinct labels for micro-expressions (MEs). The terms TP , FP , and FN correspond to the numbers of true positives, false positives, and false negatives, respectively.

Table 3 shows the results for the three-class and five-class experiments on the three datasets, with the main evaluation metrics being accuracy, UF1, and UAR.

Table 3. Results.

	Accuracy	UF1	UAR
SAMM (3-class)	86.15	78.71	78.51
CASMEII (3-class)	93.96	92.42	90.95
SMIC (3-class)	81.17	81.14	81.89
SAMM (5-class)	84.31	76.09	74.56
CASMEII (5-class)	81.63	77.34	76.78

Experiments were conducted utilizing a 2060 GPU equipped with 8GB of memory and an i7 CPU operating at 2.3GHz. The mean duration to process an image—including dataset retrieval, facial detection, and classification result generation—was 0.743 s. In contrast, the time solely dedicated to micro-expression (ME) recognition was 0.702 s.

Table 4 presents a comparative analysis of the performance outcomes of our proposed method and other methodologies in the three-class experiment. We evaluated the performance of our method against a range of prevalent ME recognition approaches, including both handcrafted feature-based methods (Bi-WOOF [34], MDMO [35], and Sparse MDMO [36]) and deep learning methods (ATNet [37], LFM [38], Capsule-Net [39], RCN [40], CBAM-DPN [41], SAM-OF [42], STSTNet [14], TSN [43], and FRL-DGT [22]).

Table 4. Comparative results (bold font indicates optimal results).

Dataset	CASME II		SMIC	
	UF1	UAR	UF1	UAR
Bi-WOOF	78.05	80.26	57.27	58.29
MDMO	49.66	51.69	59.47	59.62
Sparse MDMO	69.11	68.42	70.41	70.73
ATNet	79.80	77.50	55.30	54.30
LFM	87.00	84.00	72.00	71.00
CapsuleNet	70.68	70.18	58.20	58.77
RCN	80.87	85.63	66.00	65.00
CBAM-DPN	74.84	75.16	63.16	65.61
SAM-OFM	84.90	83.90	61.70	60.10
TSNN	61.42	60.00	69.21	68.33
STSTNet	89.10	88.20	81.20	81.40
FRL-DGT	91.90	90.30	74.30	74.90
Triple-ATFME	91.30	89.95	69.26	69.22
Ours	92.42	90.95	81.14	81.89

From the information in the table, it can be deduced that in the comparative experiments conducted, the UF1 of the network model was slightly lower on the SMIC dataset compared to STSTNet, while its performance on the other datasets and related metrics was better than that of the other models.

4.2. Ablation Experiment

To evaluate the impact of the optical flow feature extraction algorithm and the model on the task of ME recognition, ablation studies were conducted. These experiments utilized the same datasets as the previous experiments, including CASME II and SMIC. The experiments were designed as five-class recognition tasks to ensure consistency and comparability of the results. Through these controlled experiments, the specific influence of the adopted methods on improving recognition performance was clearly demonstrated, as shown in Table 5.

Table 5. Ablation study results (A⇒B indicates replacing the method in A with B. The bold font indicates that the model's prediction results are optimal).

No	Method	CASME II			SMIC		
		Accuracy	UF1	UAR	Accuracy	UF1	UAR
1	Input⇒Apex	86.75	85.31	85.00	70.13	71.70	75.91
2	Loss⇒CE	78.81	74.41	79.12	68.18	69.87	74.84
3	Loss⇒ArcFace	90.73	88.70	91.50	79.22	79.59	79.52
4	Ours	93.96	92.42	90.95	81.77	81.14	81.89

In the ablation study, we made separate replacements of the input and loss function components.

In Method 1, we replaced the original optical flow (OF) input based on the ROIs with the apex frame input. It was observed that on the CASME II dataset, the accuracy, UF1, and UAR of OF were 7.21%, 7.11%, and 5.95% higher than those of the apex frame input, respectively. On the SMIC dataset, the accuracy, UF1, and UAR of ROI_OF were 11.64%, 9.44%, and 5.98% higher than those of the apex frame unit, respectively. This suggests that using optical flow feature images as input data yields better results in ME recognition tasks compared to using untreated raw data.

In Methods 2 and 3, we replaced the loss function with the CE and ArcFace losses, respectively. It was observed that, except for the UAR on the CASME II dataset, which was slightly lower than that of the ArcFace loss, the other results were better than those of the CE loss and ArcFace loss. This indicates that this method of handling data sample imbalance has a certain level of effectiveness.

5. Conclusions

This study conducted an in-depth exploration of ME recognition technology, with a particular emphasis on utilizing ROIs for processing ME video sequences. A series of experiments validated the effectiveness of the ROI-based model in enhancing recognition accuracy and speed. This research analyzed the main challenges faced by ME recognition, including the subtlety of expressions and the scarcity of dataset samples. To address these challenges, we not only proposed an ROI-based solution but also introduced a block partition module and the WAF loss, significantly improving the model's generalization capability.

Despite the positive outcomes, there is still room for improvement in the field of ME recognition within this study. In the future, we will focus on enhancing the accuracy and real-time performance of the algorithms. Additionally, given the current scale of micro-expression datasets, there is still a need for further improvement before they can be used on a larger scale. Therefore, constructing larger and higher-quality datasets remains one of the tasks in this field. The integration of algorithms with practical application scenarios will also be explored, such as developing a visual interface to improve user experience and extend the application range of ME detection.

Author Contributions: Conceptualization, P.Z.; methodology, P.Z. and J.L.; software, P.Z. and R.W.; validation, P.Z. and J.L.; investigation, P.Z. and R.W.; writing—original draft preparation, P.Z. and R.W.; writing—review and editing, J.L. and L.S.; visualization, P.Z.; supervision, J.L. and L.S.; project administration, J.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Natural Science Foundation of China (Grant Nos. 72104016 and 62471493), the Natural Science Foundation of Chongqing, China (Grant No. CSTB2023NSCQ-MSX0391), the Beijing Natural Science Foundation (Grant No. 9242003), the Natural Science Foundation of Shandong Province (Grant Nos. ZR2023LZH017, ZR2022LZH015, ZR2024MF066), the open research subject of state key laboratory of intelligent game (No. ZBKF-24-12), the Foundation of Key Laboratory of Education Informatization for Nationalities (Yunnan Normal University), the Ministry of Education (No. EIN2024C006), and the Fundamental Research Funds for the Central Universities (No. CUCAI24003).

Institutional Review Board Statement: Ethical review and approval were waived for this study due to the anonymization of the final data used by the proposed algorithm.

Informed Consent Statement: Informed consent was obtained from all subjects involved in this study.

Data Availability Statement: The data presented in this study includes the CASME II, SAMM, and SMIC datasets, which are all publicly available on the Internet.

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

References

1. Shen, X.; Wu, Q.; Fu, X. Effects of the duration of expressions on the recognition of microexpressions. *J. Zhejiang Univ. Sci.* **2012**, *13*, 221–230. [[CrossRef](#)] [[PubMed](#)]
2. Yan, W.; Wu, Q.; Liang, J.; Chen, Y.; Fu, X. How fast are the leaked facial expressions: The duration of micro-expressions. *J. Nonverbal Behav.* **2013**, *37*, 217–230. [[CrossRef](#)]

3. Widjaja, K.S.; Alamo, C.C.; Anderies; Chowanda, A. Exploring the accuracy of artificial intelligence in detecting lies through micro-expression analysis. In Proceedings of the 2023 6th International Conference on Information and Communications Technology (ICOIACT), Yogyakarta, Indonesia, 10–11 November 2023; pp. 218–223.
4. Nair, A.J.; Manohar, S.; Mittal, A. Robotic vision with micro expression analysis: Revolutionizing customer experiences in the tourism and hospitality industry. In Proceedings of the 2023 IEEE Technology & Engineering Management Conference—Asia Pacific (TEMSCON-ASPAC), Bengaluru, India, 14–16 December 2023; pp. 1–6.
5. Kim, H. Paul ekman and the search for the isolated face in the 1960s. *J. Hist. Behav. Sci.* **2024**, *60*, e22322. [[CrossRef](#)] [[PubMed](#)]
6. Friesen, E.; Ekman, P. Facial action coding system: A technique for the measurement of facial movement. *Palo Alto* **1978**, *3*, 5.
7. Ekman, P. An argument for basic emotions. *Cogn. Emot.* **1992**, *6*, 169–200. [[CrossRef](#)]
8. Ojala, T.; Pietikainen, M.; Harwood, D. Performance evaluation of texture measures with classification based on kullback discrimination of distributions. In Proceedings of the 12th International Conference on Pattern Recognition, Jerusalem, Israel, 9–13 October 1994; IEEE: Piscataway, NJ, USA, 1994; Volume 1, pp. 582–585.
9. Zong, Y.; Huang, X.; Zheng, W.; Cui, Z.; Zhao, G. Learning from hierarchical spatiotemporal descriptors for micro-expression recognition. *IEEE Trans. Multimed.* **2018**, *20*, 3160–3172. [[CrossRef](#)]
10. Bahmanabady, E.; Imani, M.; Ghassemian, H. Fusion of multi-level cnn with lbp features for facial emotion recognition. In Proceedings of the 2023 31st International Conference on Electrical Engineering (ICEE), Tehran, Iran, 9–11 May 2023; pp. 463–467.
11. Shreve, M.; Godavarthy, S.; Goldgof, D.; Sarkar, S. Macro-and micro-expression spotting in long videos using spatio-temporal strain. In Proceedings of the 2011 IEEE International Conference on Automatic Face & Gesture Recognition (FG), Santa Barbara, CA, USA, 21–23 March 2011; IEEE: Piscataway, NJ, USA, 2011; pp. 51–56.
12. Li, Y.; Wei, J.; Liu, Y.; Kauttonen, J.; Zhao, G. Deep learning for micro-expression recognition: A survey. *IEEE Trans. Affect. Comput.* **2022**, *13*, 2028–2046. [[CrossRef](#)]
13. Kim, D.H.; Baddar, W.J.; Ro, Y.M. Micro-expression recognition with expression-state constrained spatio-temporal feature representations. In Proceedings of the 24th ACM International Conference on Multimedia, Amsterdam, The Netherlands, 15–19 October 2016; pp. 382–386.
14. Liong, S.; Gan, Y.S.; See, J.; Khor, H.; Huang, Y. Shallow triple stream three-dimensional cnn (ststnet) for micro-expression recognition. In Proceedings of the 2019 14th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2019), Lille, France, 14–18 May 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 1–5.
15. Zhang, L.; Hong, X.; Arandjelović, O.; Zhao, G. Short and long range relation based spatio-temporal transformer for micro-expression recognition. *IEEE Trans. Affect. Comput.* **2022**, *13*, 1973–1985. [[CrossRef](#)]
16. Wang, C.; Peng, M.; Bi, T.; Chen, T. Micro-attention for micro-expression recognition. *Neurocomputing* **2020**, *410*, 354–362. [[CrossRef](#)]
17. Xia, Z.; Hong, X.; Gao, X.; Feng, X.; Zhao, G. Spatiotemporal recurrent convolutional networks for recognizing spontaneous micro-expressions. *IEEE Trans. Multimed.* **2019**, *22*, 626–640. [[CrossRef](#)]
18. Peng, M.; Wu, Z.; Zhang, Z.; Chen, T. From macro to micro expression recognition: Deep learning on small datasets using transfer learning. In Proceedings of the 2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018), Xi'an, China, 15–19 May 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 657–661.
19. Li, F.; Nie, P.; You, M.; Chen, Z.; Wang, G. Triple-atfme: Triple-branch attention fusion network for micro-expression recognition. *Arab. J. Sci. Eng.* **2024**, 1–17. [[CrossRef](#)]
20. Wang, Z.; Zhang, K.; Luo, W.; Sankaranarayana, R. Htnet for micro-expression recognition. *Neurocomputing* **2024**, *602*, 128196. [[CrossRef](#)]
21. Fan, X.; Chen, X.; Jiang, M.; Shahid, A.R.; Yan, H. Selfme: Self-supervised motion learning for micro-expression recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; pp. 13834–13843.
22. Zhai, Z.; Zhao, J.; Long, C.; Xu, W.; He, S.; Zhao, H. Feature representation learning with adaptive displacement generation and transformer fusion for micro-expression recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; pp. 22086–22095.
23. Tinhinane, M.; Yassine, N.; Ahmad, A.; Soraya, A.; Rachid, J. Effects of region of interest location on osteoarthritis detection using deep feature learning. In Proceedings of the 2023 Twelfth International Conference on Image Processing Theory, Tools and Applications (IPTA), Paris, France, 16–19 October 2023; IEEE: Piscataway, NJ, USA, 2023; pp. 1–6.
24. Zhang, T.; Zong, Y.; Zheng, W.; Chen, C.L.P.; Hong, X.; Tang, C.; Cui, Z.; Zhao, G. Cross-database micro-expression recognition: A benchmark. *IEEE Trans. Knowl. Data Eng.* **2022**, *34*, 544–559. [[CrossRef](#)]
25. Yan, W.; Li, X.; Wang, S.; Zhao, G.; Liu, Y.; Chen, Y.; Fu, X. Casme ii: An improved spontaneous micro-expression database and the baseline evaluation. *PLoS ONE* **2014**, *9*, e86041. [[CrossRef](#)]

26. Liong, S.; See, J.; Phan, R.C.; Ngo, A.C.L.; Oh, Y.; Wong, K. Subtle expression recognition using optical strain weighted features. In Proceedings of the Computer Vision-ACCV 2014 Workshops, Singapore, 1–2 November 2014; Revised Selected Papers, Part II 12; Springer: Berlin/Heidelberg, Germany, 2015; pp. 644–657.
27. Happy, S.; Routray, A. Fuzzy histogram of optical flow orientations for micro-expression recognition. *IEEE Trans. Affect. Comput.* **2017**, *10*, 394–406. [[CrossRef](#)]
28. Zach, C.; Pock, T.; Bischof, H. A duality based approach for realtime tv-l 1 optical flow. In Proceedings of the Pattern Recognition: 29th DAGM Symposium, Heidelberg, Germany, 12–14 September 2007; Proceedings 29; Springer: Berlin/Heidelberg, Germany, 2007; pp. 214–223.
29. Liu, Y.; Du, H.; Zheng, L.; Gedeon, T. A neural micro-expression recognizer. In Proceedings of the 2019 14th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2019), Lille, France, 14–18 May 2019; pp. 1–4.
30. Deng, J.; Guo, J.; Xue, N.; Zafeiriou, S. Arcface: Additive angular margin loss for deep face recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 4690–4699.
31. Li, X.; Pfister, T.; Huang, X.; Zhao, G.; Pietikäinen, M. A spontaneous micro-expression database: Inducement, collection and baseline. In Proceedings of the 2013 10th IEEE International Conference and Workshops on Automatic face and Gesture Recognition (fg), Shanghai, China, 22–26 April 2013; IEEE: Piscataway, NJ, USA, 2013; pp. 1–6.
32. Yap, C.H.; Kendrick, C.; Yap, M.H. Samm long videos: A spontaneous facial micro-and macro-expressions dataset. In Proceedings of the 2020 15th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2020), Buenos Aires, Argentina, 16–20 November 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 771–776.
33. John, S.; Yap, M.H.; Li, J.; Hong, X.; Wang, S. MEGC 2019—The second facial micro-expressions grand challenge. In Proceedings of the 2019 14th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2019), Lille, France, 14–18 May 2019; pp. 1–5.
34. Liong, S.; See, J.; Wong, K.; Phan, R.C. Less is more: Micro-expression recognition from video using apex frame. *Signal Process. Image Commun.* **2018**, *62*, 82–92. [[CrossRef](#)]
35. Liu, Y.; Zhang, J.; Yan, W.; Wang, S.; Zhao, G.; Fu, X. A main directional mean optical flow feature for spontaneous micro-expression recognition. *IEEE Trans. Affect. Comput.* **2015**, *7*, 299–310. [[CrossRef](#)]
36. Liu, Y.; Li, B.; Lai, Y. Sparse mdmo: Learning a discriminative feature for micro-expression recognition. *IEEE Trans. Affect. Comput.* **2018**, *12*, 254–261. [[CrossRef](#)]
37. Peng, M.; Wang, C.; Bi, T.; Shi, Y.; Zhou, X.; Chen, T. A novel apex-time network for cross-dataset micro-expression recognition. In Proceedings of the 2019 8th International Conference on Affective Computing and Intelligent Interaction (ACII), Cambridge, UK, 3–6 September 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 1–6.
38. Choi, D.Y.; Song, B.C. Facial micro-expression recognition using two-dimensional landmark feature maps. *IEEE Access* **2020**, *8*, 121549–121563. [[CrossRef](#)]
39. Quang, N.V.; Chun, J.; Tokuyama, T. Capsulenet for micro-expression recognition. In Proceedings of the 2019 14th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2019), Lille, France, 14–18 May 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 1–7.
40. Xia, Z.; Peng, W.; Khor, H.; Feng, X.; Zhao, G. Revealing the invisible with model and data shrinking for composite-database micro-expression recognition. *IEEE Trans. Image Process.* **2020**, *29*, 8590–8605. [[CrossRef](#)] [[PubMed](#)]
41. Niu, R.; Yang, J.; Xing, L.; Wu, R. Micro-expression recognition algorithm based on convolutional block attention module and dual path networks. *J. Comput. Appl.* **2021**, *41*, 2552.
42. Dezhi, L.; Zhengyou, L.; Yu, S. Micro-expression recognition method based on spatial attention mechanism and optical flow features. *J.-Comput.-Aided Des. Comput. Graph.* **2021**, *33*, 1541–1552.
43. Wu, C.; Guo, F. Tsnn: Three-stream combining 2d and 3d convolutional neural network for micro-expression recognition. *IEEE Trans. Electr. Electron. Eng.* **2021**, *16*, 98–107. [[CrossRef](#)]

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