Electronic Research Archive

ERA, 32(4): 2745−2771.
DOI: 10.3934/era.2024124
Received: 31 January 2024
Revised: 01 March 2024
Accepted: 14 March 2024
Published: 03 April 2024

https://www.aimspress.com/journal/ERA

Research article

Incorporating eyebrow and eye state information for facial expression recognition in mask-obscured scenes

Kun Zheng[1], Li Tian[1], Zichong Li[1], Hui Li[1,*] and Junjie Zhang[2,*]

[1] Faculty of Information Technology, Beijing University of Technology, Beijing 100124, China
[2] Smart Learning Institute, Beijing Normal University, Beijing 100875, China

* **Correspondence:** Email: lihui@bjut.edu.cn, 11132022024@bnu.edu.cn; Tel: +8618810899182, +8618911610204.

Abstract: Facial expression recognition plays a crucial role in human-computer intelligent interaction. Due to the problem of missing facial information caused by face masks, the average accuracy of facial expression recognition algorithms in mask-obscured scenes is relatively low. At present, most deep learning-based facial expression recognition methods primarily focus on global facial features, thus they are less suitable for scenarios where facial expressions are obscured by masks. Therefore, this paper proposes a facial expression recognition method, TransformerKNN (TKNN), which integrates eyebrow and eye state information in mask-obscured scenes. The proposed method utilizes facial feature points in the eyebrow and eye regions to calculate various relative distances and angles, capturing the state information of eyebrows and eyes. Subsequently, the original face images with masks are used to train a Swin-transformer model, and the eyebrow and eye state information is used to train a k-Nearest Neighbor (KNN) model. These models are then fused at the decision layer to achieve automated emotion computation in situations when facial expressions are obscured by masks. The TKNN method offers a novel approach by leveraging both local and global facial features, thereby enhancing the performance of facial expression recognition in mask-obscured scenes. Experimental results demonstrate that the average accuracy of the TKNN method is 85.8% and 70.3%, respectively. This provides better support for facial expression recognition in scenarios when facial information is partially obscured.

Keywords: facial expression recognition; mask-obscured; eyebrow and eye state; local and global facial features

1. Introduction

Expression is a crucial channel for conveying emotional information and reflecting the emotional state of humans [1]. Facial expression recognition has been widely applied in human-computer interaction fields, such as intelligent education, intelligent driving, and medical diagnosis. However, this technology encounters numerous challenges, including occlusion, changes in facial posture, variations in lighting conditions, head movements, and individual differences among individuals. Despite existing facial expression recognition methods are effective in addressing many challenges, further research is needed to recognize facial expressions in occluded scenarios [2].

Kotsia et al. [3] revealed that occlusion of the mouth region results in the most significant loss of facial expression information for facial expression recognition by separately occluding the mouth, eyes, and left-right facial regions. However, in the context of the ongoing pandemic, face masks have become an integral part of people's daily lives. The occlusion of the mouth and nose areas by masks is expected to have a negative impact on the accuracy of facial expression recognition. Wong and Estudillo [4] found that wearing face masks resulted in a decrease in the average accuracy of human subjective facial expression recognition from 61.8% to 45.0%, and Cooper et al. [5] indicated a drop from 75.4% to 52.6%. In Grundmann et al.'s study [6], the accuracy of facial expression recognition decreased from 69.9% to 48.9%. Marini et al.'s study [7] also confirmed the impact of masks on emotion recognition, and indicated that the influence of masks on different expression types varied. In this case, improving the accuracy of facial expression recognition under occlusion conditions is crucial.

The analysis of facial expression recognition under occlusion conditions is conducted based on methods and the selection of data regions.

(1) Based on methods

From the perspective of methods, facial expression recognition methods under occlusion conditions can be categorized into traditional methods, deep learning methods, and hybrid methods combining traditional and deep learning approaches.

Traditional methods are known for their simplicity and fast processing speed. Zhang et al. [8] proposed a method for handling facial expression images under partial occlusion using template matching. They applied Gabor filters to the images, extracted local facial templates using the Monte Carlo algorithm, used template matching to generate robust features against occlusion, and then trained a Support Vector Machine with these features. Experimental results showed that this method achieved testing accuracies of 90.8% and 78.4% on the CK+ [9] and JAFFE [10] datasets under mouth occlusion, respectively. However, traditional methods usually lack sufficient discriminative capability and have poor generalization performance.

Deep learning methods automatically learn expressive features from images with partial occlusion. Ding et al. [11] introduced an occlusion-adaptive deep network with two branches. The attention branch based on feature points guides the network to focus on non-occluded facial regions, while the region-based branch divides the feature map into non-overlapping facial blocks for separate classifier training. This method achieved accuracies of 84.6% and 64.0% on the occluded FERPlus [12] and AffectNet [13] datasets, respectively. Wang et al. [14] utilized region attention networks to enhance attention to specific regions, divided input images into multiple regions, and calculated the contribution of each region to the expression recognition task. The method achieved accuracies of 83.6%, 58.5%, and 82.7% on FERPlus, AffectNet, and RAF-DB [15], respectively. Deep learning methods can automatically learn abstract and high-level features from input data without the need for manual feature extractor design. Although deep

learning methods have achieved significant results in solving facial expression recognition, these methods often require substantial computational resources and training time.

Hybrid methods combining traditional and deep learning approaches have the advantages of leveraging the speed and simplicity of traditional methods, while improving robustness and generalization abilities of the model. Dapogny et al. [16] proposed the confidence-weighted local expression predictions. This method introduced a Random Forest for learning facial local subspaces, and utilized a hierarchical autoencoder network to calculate confidence in each local subspace. It achieved an accuracy of 72.7% under mouth occlusion on the CK+ dataset, and demonstrated excellent robustness to facial occlusion. However, research on facial expression recognition under mask occlusion is limited, and there is room for improvement in accuracy. In this study, we choose to use a Transformer network to train a global facial expression feature model. The transformer models, known for their self-attention and multi-head attention mechanisms, have been widely applied in natural language processing and computer vision tasks, including models such as vision-transformer [17] and Swin-transformer [18]. Compared with Vision-transformer, Swin-transformer introduces window-based self-attention calculations, reducing computational requirements and training time. However, the average accuracy of expression recognition based on Swin-transformer needs to be further improved under occlusion conditions.

(2) Based on selection of data regions

From the perspective of selection of data regions, facial expression recognition methods under occlusion conditions can be categorized into two types: methods of reconstructing occluded regions and methods of emphasizing unoccluded regions.

The methods of reconstructing occluded regions aim to return the network regress to an ideal situation which can recognize the entire facial expression. The pioneering research [19] proposed calculating optical flow from two occluded face frames and using an autoencoder to recover occluded optical flow for predicting facial expressions. Lou et al. [20] utilized generative adversarial network technology to combine a facial expression classifier with a 3D facial model, achieving realistic facial expression reconstruction for virtual reality head-mounted display users. Although methods of reconstructing occluded regions have made some progress in addressing occlusion issues, these approaches lack authenticity in restoring facial expression details and exhibit limited generalization abilities across different scenes or individuals.

The methods of emphasizing unoccluded regions are based on the idea that humans can quickly focus on interesting objects in complex visual scenes [21]. Li et al. [22] utilized gate units to calculate adaptive weights, and controlled the flow of information in a convolutional neural network based on the occlusion status and importance of the facial region of interest. Liu et al. [23] proposed a robust regularization encoding, which assigned weights to each pixel in the image and iteratively calculated weights through regular regression coefficients until a threshold was found to reduce the weights of occluded pixels. Emphasizing unoccluded regions allows the network to undergo more targeted training, aiding in capturing key features more accurately. Ekman et al. [24] proposed dividing the face into several Action Units (AUs) to describe facial expressions. The Action Units occluded by mask are shown in Table 1, and the Action Units unoccluded by mask are shown in Table 2. It is obvious that the unoccluded region, mainly including the eyebrow and eye areas, is significant. Therefore, we consider adopting the method of emphasizing unoccluded region, especifically the eyebrow and eye regions.

Table 1. Action Units occluded by mask.

AU	Meaning	AU	Meaning
AU6	Lifting cheek and tightening the outer orbicularis oculi muscle	AU24	Pressing the lips against each other
AU8	Facing lips to each other	AU25	Separating the lips to expose the teeth
AU9	Wrinkling nose	AU26	Separating the lips to expose the tongue
AU10	Pulling the upper lip upwards	AU27	Separating the lips to expose the throat
AU11	Pulling the skin in the philtrum area upwards	AU28	Sucking lips
AU12	Pulling the corners of the mouth upwards at an angle	AD29	Pushing the chin downwards
AU13	Rapid movement of the lips	AD30	Moving the jaw to the left or right
AU14	Tightening the lips	AU31	Clamping the jaw tightly
AU15	Pulling the corners of the mouth downwards at an angle	AD32	Biting the lip
AU16	Pulling the lower lip downwards	AD33	Blowing air
AU17	Pulling the lower lip upwards	AD34	Inflating cheeks
AU18	Pouting.	AD35	Sucking
AD19	Sticking out the tongue	AD36	Tongue hitting the cheek
AU20	Stretching the corners of the mouth	AD37	Tongue licking the lips
AU22	Tightening the lips and turning them outward	AD38	Flaring nasal
AU23	Tightening the lips	AU39	Constricting nasal

Table 2. Action Units unoccluded by mask.

AU	Meaning	AU	Meaning
AU1	Raising the inner corner of the eyebrows	AU41	Drooping upper eyelids slightly
AU2	Lifting the outer corner of the eyebrows	AU42	Drooping upper eyelids
AU4	Frowning or wrinkling the brow	AU43	Frowning or wrinkling the brow
AU5	Rising upper eyelids	AU44	Pushing the lower eyelids upwards
AU7	Tightening of the inner circle of the orbicularis oculi muscle	AU45	Blinking
AU21	Tensing the neck	AU46	Darting eyes

The eyebrow and eye region have been widely utilized in various fields in previous researches. In the field of face recognition, Ramachandra and Ramachandran [25] employed the KAZE [26], HOG [27], and SING (sub-image-based neighbor gradient feature extraction) algorithms to extract features from the eyebrow and eye regions. They estimated the eyebrow shape features based on the width, height of the eyebrows, and the distance from N points in the eyebrow region to the corner point of the eye,

achieving robust periocular recognition. Huang et al. [28] introduced a face recognition model based on the graph convolutional network (GCN). The model utilizes the symmetric similarity between the left and right eyebrows, as well as the subordinate relationship between facial components and the entire face, in order to achieve face recognition. In the field of heart rate detection, Zheng et al. [29] used the VJ eye detector [30] to detect eyes and selected the forehead as the region of interest, in order to complete heart rate detection in the absence of facial information. In addition, the eyebrow and eye regions also have applications in the field of emotion recognition. Li et al. [31] demonstrated the correlation between different eyebrow contour features and facial expressions, while Zhang et al. [32] proposed that the state of the eyes can reflect changes in facial expressions. However, the previous methods may not be sensitive enough to subtle changes in facial expressions, and they are not specifically designed for recognition under mask-obscured scenes, leading to poor performance in recognizing facial expressions. Table 2 indicates that multiple Action Units near the eyebrows and eyes are not obscured by masks, providing discriminative cues for expression recognition under mask occlusion. For example, AU1 may indicate concern or displeasure, AU2 may indicate surprise or happiness, AU4 and AU6 may represent anger, and AU7 may suggest surprise. We propose utilizing visible landmarks outside the mask to estimate detailed information about the eyebrows and eyes, achieving more precise feature extraction.

Combining global and local information enables obtaining more comprehensive information, enhancing system performance and robustness, as well as strengthening feature representation capabilities, thereby better accomplishing various tasks. Based on this idea, Tao, et al. [33] introduced the detail-difference-aware module, which prioritized the most informative visual elements in the spatial domain and an attention-based feature separation module, reducing the interference of background information on smoke information. Through the multiconnection aggregation method, local and global features were fully aggregated, ultimately achieving precise identification of forest smoke. Similarly, the hierarchical attention network with progressive feature fusion [34] enhanced the focus on features relevant to expression information and suppressed the interference of irrelevant features through a diverse feature extraction module, aggregating complementary features of local and global contexts, as well as spatial gradient features.

In summary, we follow the approach of integrating global facial and local eyebrow-eye information. We introduce a method named transformerKNN (TKNN), which combines a Swin-transformer-based Model-a and a KNN-based Model-b. The key steps are illustrated in Figure 1. Through the Swin-transformer network, the model can extract facial expression features based on the entire face, emphasizing global attention. The KNN module is used to classify expressions around the eyebrows and eyes, providing local attention for the model. The specific contributions of this paper are as follows:

(1) A deep learning-based expression recognition method is proposed to address the issue of low average accuracy in expression recognition under mask occlusion. The algorithm combines eyebrow and eye state in mask-obscured scenarios. The Swin-transformer network and the KNN model, which incorporates information regarding eyebrow and eye states, are fused together at the decision-making layer, enabling the model to focus on the unoccluded regions, and improving the average accuracy of expression recognition.

(2) In response to the notable scarcity of expression datasets capturing individuals wearing masks, we have taken proactive steps to address this issue by creating expression datasets under mask-obscured scenes utilizing open-source tools. As a result, we have obtained the datasets MYRAF-

3_KN95, MYLFW_KN95, and RAF_KN95. These datasets are distinguished by their meticulous masking of facial features, thereby guaranteeing heightened robustness and presenting a more natural appearance.

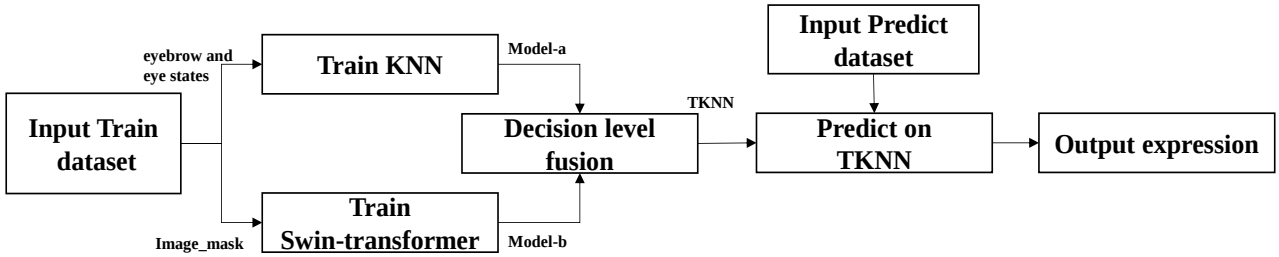


Figure 1. The method for TKNN.

The organizational structure of this paper is as follows. Section 2 introduces the proposed expression recognition method that incorporating eyebrow and eye state information for facial expression recognition in mask-obscured scenes. Emphasis is placed on the state information around the eyebrows and eyes. The model construction involves the introduction of a new model named TKNN, which combines Swin-transformer and KNN at the decision level. The purpose is to improve the average accuracy of expression recognition with Swin-transformer by incorporating eyebrow and eye state information. Section 3 explores the distribution of eyebrow and eye state, expression recognition results of Swin-transformer method, fusion results of different machine learning models and Swin-transformer, expression recognition results with introduced eyebrow and eye feature points, and validates the classification performance of our method on three-class and seven-class masked facial expression datasets. Section 4 concludes the paper and suggests further improvements.

2. Incorporating eyebrow and eye state information for facial expression recognition in mask-obscured scenes

Our goal is to achieve accurate expression recognition in scenes where facial features are obscured by masks. The system structure diagram of our method is shown in Figure 2. Initially, we utilize the open-source tool MaskTheFace [35] to generate masks for the facial expression datasets. Subsequently, the masked datasets are fed into the Swin-transformer for training, to obtain Model-a. Concurrently, we utilize feature point detection technology to generate facial feature points, especially around the eyes and eyebrows, to calculate eye and eyebrow state information. This information is then input into KNN for training, in order to develop Model-b. During the final prediction process, our approach utilizes Model-b to correct the false detections of Model-a, achieving decision-level fusion, namely TKNN. The following sections will provide a detailed description of our approach.

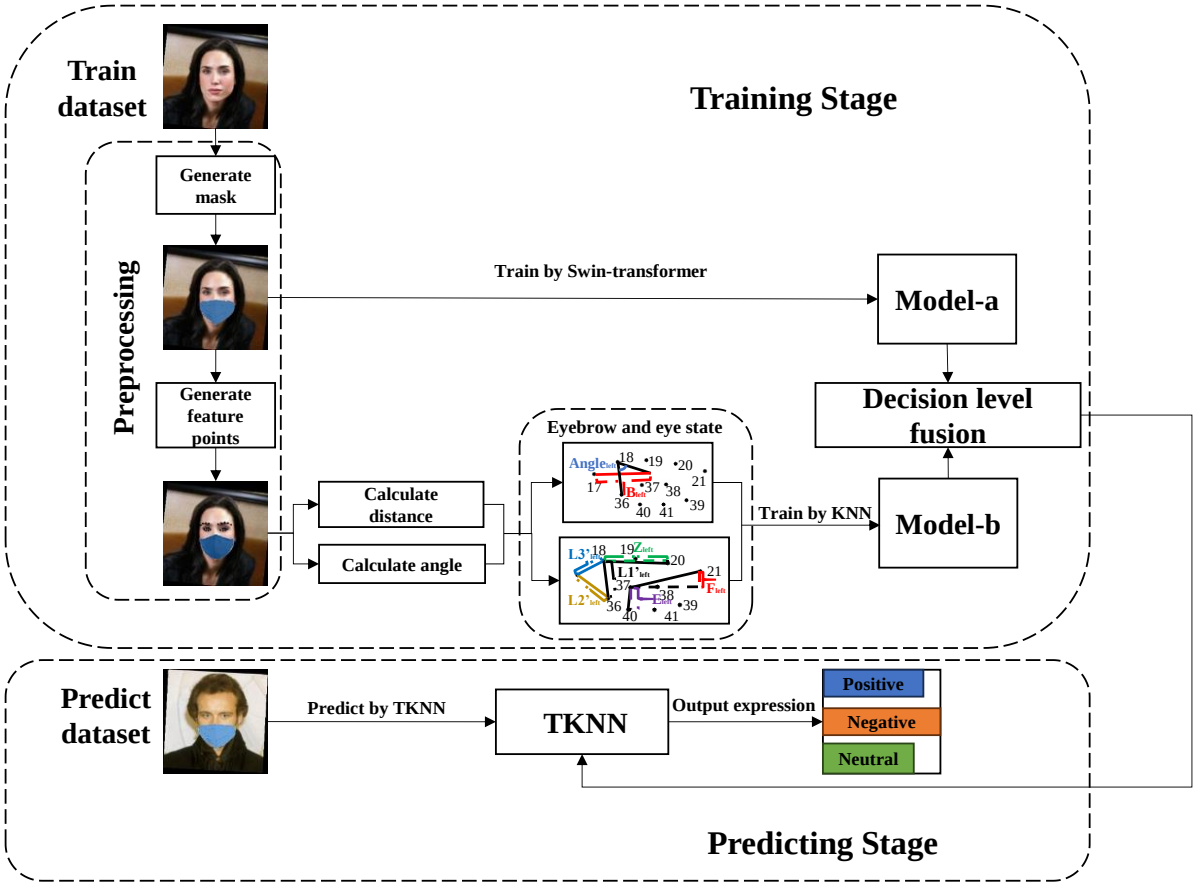


Figure 2. System structure diagram of the TKNN method.

2.1. Image preprocessing

2.1.1. Generate mask

Due to the absence of expression datasets under mask-obscured scenes, we contemplate the utilization of the open-source tool MaskTheFace to add masks to the existing datasets. As illustrated in Figure 3, we can flexibly create masks of diverse shapes and colors, each of which can result in varying degrees of facial obstruction. In this context, we opt for the mask type (e), which provides maximal facial coverage, and label it as the KN95 mask. Specifically, we use the dlib.get_frontal_face_detector function from the Dlib library to establish a face detector [36], which utilizes techniques, such as HOG, linear classifiers, pyramid image structures, and sliding window detection to identify facial bounding boxes. We utilize the dlib.shape_predictor function as the feature point detection model to generate 68 facial feature points and estimate the positions of the six key locations of the mask on the face. The facial tilt angle is calculated, and based on this angle, we select the template for the KN95 mask with different orientations. Individuals wearing masks facing in various directions are shown in Figure 4. This mask generation method exhibits robustness, capable of adapting well to head tilt angles, effectively covering the nose and mouth, while presenting a natural effect.

(a) (b) (c) (d) (e) (f) (g) (h) (i)

Figure 3. Pictures of different mask types.

(a) (b) (c)

Figure 4. Examples of individuals wearing masks in different orientations. (a) Front-facing. (b) Left-facing. (c) Right-facing.

RAF-DB [15] is a large-scale dataset focused on facial expressions. It contains 29,672 diverse facial expression images, labeled by 40 annotators for basic or compound expressions. We discuss only the basic expressions, so only seven types of basic expression images are selected, including surprise, fear, disgust, happy, sad, angry, and neutral, with a total of 15,339 images. Then, these 15,339 images are divided into a new Training set and a validation set in a 9 : 1 ratio. To simplify the model training and evaluation process and enhance its generalization ability, the dataset is further categorized into positive, negative, and neutral, named RAF-3 dataset. Specifically, surprise, fear, disgust, sad, and angry belong to the negative category, happy belongs to the positive category, and neutral belongs to the neutral category. This classification method is more intuitive and easier to understand, while also aligning better with real-world emotional expressions. Additionally, this simplified classification approach reduces experimental complexity and improves efficiency. Due to the process of adding masks, detection failures may occur when analyzing images in some cases. Consequently, the number of images decreases to 9838, named RAF-3_KN95 dataset. We detect images with detectable eye and eyebrow state in both RAF-3 dataset and RAF-3_KN95 dataset, renaming them as MYRAF-3 and MYRAF-3_KN95 datasets. For the RAF-3_KN95 dataset and MYRAF-3_KN95 dataset, we generate facial feature points 17−26 and 36−47 at the eyes and eyebrows to obtain RAF-3_KN95tzd dataset and MYRAF-3_KN95tzd dataset by feature point detection model, as detailed in Section 2.1.2.

The LFW-FER dataset [37] is a derivative facial expression dataset based on the LFW face dataset [38]. It selects 10,487 images from the 13,000 samples in the LFW dataset and manually annotates them into positive, negative, and neutral categories. We consider only single-person images after adding masks, totaling 10,019 images. The Training set is split into a new Training set and a Test set in a 9 : 1 ratio, named LFW-FER_KN95 dataset. We detect images with detectable eye and eyebrow state in both LFW_FER dataset and LFW_FER_KN95 dataset, renaming them as MYLFW_FER and MYLFW_FER_KN95 datasets. For the LFW_FER_KN95 dataset and MYLFW_FER_KN95 dataset, we generate facial feature points 17−26 and 36−47 at the eyes and eyebrows, obtaining

LFW_FER_KN95tzd dataset and MYLFW_FER_KN95tzd dataset.

2.1.2. Generate feature points

The feature point detection model commonly used to generate facial feature points include MTCNN [39] and Dlib. MTCNN utilizes a cascaded convolutional neural network with three stages (P-Net, R-Net, O-Net) to progressively refine the prediction of five key facial points: Both eyes, the tip of the nose, and the corners of the mouth, as shown in Figure 5(b). After wearing a mask, the feature points around the tip of the nose and corners of the mouth are obscured, as depicted in Figure 5(c). Dlib typically uses lines or circles to mark 68 facial landmarks, covering areas such as eyebrows, nose bridge, tip of the nose, eyes, lips, and cheeks, as shown in Figure 6(b). Facial feature points after wearing a mask are shown in Figure 6(c), indicating that points around the nose bridge, tip of the nose, and lips (points 29−35 and 48−67 are likely to be obscured, while feature points around the eyebrows and eyes are retained. Therefore, even when wearing a KN95 mask, a series of points on the brow center, brow ridge, brow tail, eye contour, as well as feature points on the upper and lower eyelids, can be detected.

(a) (b) (c)

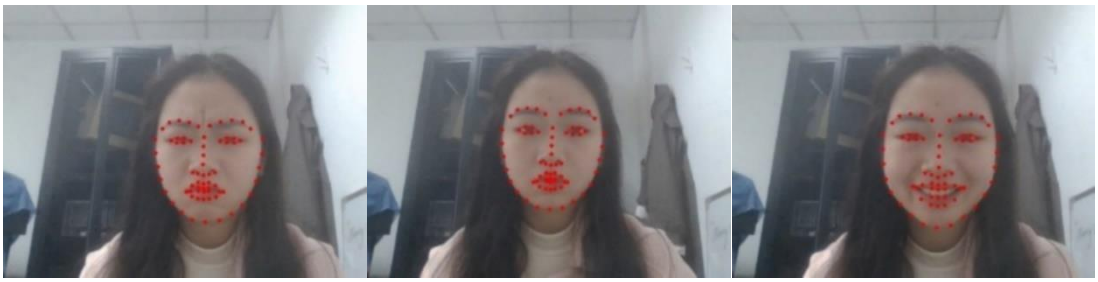
Figure 5. Genarate feature points based on MTCNN. (a) Original image. (b) Image with 5 feature points generated on MTCNN. (c) Image with feature points after wearing a KN95 mask.

(a) (b) (c)

Figure 6. Genarate feature points based on Dlib. (a) Original image. (b) Image with 68 feature points generated on Dlib. (c) Image with feature points after wearing a KN95 mask.

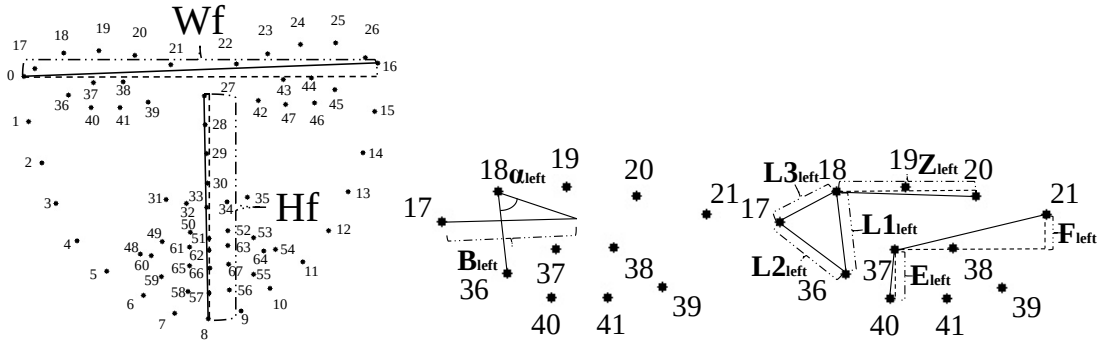
2.2. Obtain eye and eyebrow state

We use the Dlib face detector to detect facial feature points for positive, neutral, and negative expressions, as shown in Figure 7. It can be observed that when facial expressions change, feature points around the eyebrows, eyes, and mouth undergo noticeable variations, leading to changes in the distances between each feature point. As explained in Section 2.1, compare to normal facial images, facial images in mask-obscured scenarios have limited visible feature points. Therefore, it is crucial to make full use of the features around the eyes and eyebrows. We utilize Dlib as feature point detection model to detect feature points around the eyes and eyebrows, inferring the morphology of the eyebrows and eyes, as well as the relative positional information between the eyes and eyebrows, as illustrated in Figure 8.



(a) (b) (c)

Figure 7. Variation of Dlib feature points with facial expressions. (a) Positive, (b) neutral, and (c) negative.



(a) (b) (c)

Figure 8. The acquisition of eye and eyebrow state. (a) Facial width and facial height. (b) and (c) Partial eye and eyebrow state features on the left eye.

Specifically, the feature point detection model detects the height and width of the facial bounding box H and W, the degree of eye openness He, the degree of eyebrow raising Hb, the degree of eyebrow furrowing Wb, the relative proportional relationship between the distance from the eye corner to the eyebrow ridge and the facial width $L1$, the relative proportional relationship between the distance from the eye corner to the eyebrow tip and the facial width $L2$, the relative proportional relationship between the distance from the eyebrow tip to the eyebrow tail and the facial width $L3$, the angle between the

midpoint of the eyebrow and the eyebrow ridge and the distance from the eye corner to the eyebrow ridge *Angle*, and the ratio between the distance from the midpoint of the eyebrow to the eyebrow tail and the distance from the eyebrow ridge to the eyebrow tip *Ratio*. The calculation formulas are shown in Eqs (1)–(14).

$$H = dy_{27} - dy_8 \tag{1}$$

$$W = dx_{16} - dx_0 \tag{2}$$

$$He = \frac{E_{left}+E_{right}}{2\times H} = \frac{|dy_{37}-dy_{40}|+|dy_3-dy_{47}|}{2\times H} \tag{3}$$

$$Hb = \frac{F_{left}+F_{right}}{2\times H} = \frac{|dy_{21}-dy_{37}|+|dy_{22}-dy_{44}|}{2\times H} \tag{4}$$

$$Wb = \frac{Z_{left}+Z_{right}}{2\times W} = \frac{|dx_{18}-dx_{20}|+|dx_{23}-dx_{25}|}{2\times W} \tag{5}$$

$$L1 = \frac{L1'}{W} = \frac{L1'_{left}+L1'_{right}}{2\times W} = \frac{\sqrt{(dy_{36}-dy_{18})^2-(dx_{36}-dx_{18})^2}+\sqrt{(dy_{45}-dy_{25})^2-(dx_{45}-dx_{25})^2}}{2\times W} \tag{6}$$

$$L2 = \frac{L2'}{W} = \frac{L2'_{left}+L2'_{right}}{2\times W} = \frac{\sqrt{(dy_{36}-dy_{17})^2-(dx_{36}-dx_{17})^2}+\sqrt{(dy_{45}-dy_{26})^2-(dx_{45}-dx_{26})^2}}{2\times W} \tag{7}$$

$$L3 = \frac{L3'}{W} = \frac{L3'_{left}+L3'_{right}}{2\times W} = \frac{\sqrt{(dy_{18}-dy_{17})^2-(dx_{18}-dx_{17})^2}+\sqrt{(dy_{26}-dy_{25})^2-(dx_{26}-dx_{25})^2}}{2\times W} \tag{8}$$

$$Dbrow_{left} = \begin{pmatrix} \frac{dy_{17}+dy_{21}}{2} & dy_{18} \\ \frac{dx_{17}+dx_{21}}{2} & dx_{18} \end{pmatrix} \tag{9}$$

$$Dbrow_{right} = \begin{pmatrix} \frac{dy_{22}+dy_{26}}{2} & dy_{25} \\ \frac{dx_{22}+dx_{26}}{2} & dx_{25} \end{pmatrix} \tag{10}$$

$$Deye_{left} = \begin{pmatrix} dy_{36} & dy_{18} \\ dx_{36} & dx_{18} \end{pmatrix} \tag{11}$$

$$Deye_{right} = \begin{pmatrix} dy_{45} & dy_{25} \\ dx_{45} & dx_{25} \end{pmatrix} \tag{12}$$

$$Angle = \frac{Angle_{left}+Angle_{right}}{2} = \frac{\frac{1}{\cos\left(\frac{Dbrow_{left}\cdot Deye_{left}}{\|Dbrow_{left}\|\cdot L1_{left}}\right)}\cdot\frac{180}{\pi}+\frac{1}{\cos\left(\frac{Dbrow_{right}\cdot Deye_{right}}{\|Dbrow_{right}\|\cdot L1_{right}}\right)}\cdot\frac{180}{\pi}}{2} \tag{13}$$

$$Ratio = \frac{B_{left}+B_{right}}{2\times L3} = \frac{\sqrt{\left(\frac{dy_{17}+dy_{21}}{2}-dy_{21}\right)^2}+\sqrt{\left(\frac{dx_{17}+dx_{21}}{2}-dx_{21}\right)^2}+\sqrt{\left(\frac{dy_{22}+dy_{26}}{2}-dy_{26}\right)^2}+\sqrt{\left(\frac{dx_{22}+dx_{26}}{2}-dx_{26}\right)^2}}{2\times L3} \tag{14}$$

Here, dy_i and dx_i respectively represent the vertical and horizontal coordinates of the i-th feature point, where i ranges from 0 to 67. *Dbrow* is vector from the midpoint of the eyebrow to the eyebrow

ridge, *Deye* is vector from the eye corner to the eyebrow ridge. ||*Dbrow*|| represents the magnitudes of the vector *Dbrow*. *B* is vector from the midpoint of the eyebrow to the eyebrow tip. It is important to note that due to differences in datasets and facial sizes, the coordinates of the feature points obtained by feature point detection model may change. Therefore, the relative positional relationship between the feature point coordinates and the facial bounding box is used to represent the status features around the eyes and eyebrows.

2.3. Train model

2.3.1. Model-a based on Swin-transformer

We choose Swin-transformer as the primary deep learning framework to learn the facial expression features of images with individuals wearing masks using preprocessed data from Section 2.1. The structure of Swin-transformer is shown in Figure 9. Specifically, the images are resized to 224 × 224 × 3, and divided into non-overlapping local regions of size 4 × 4 using the Patch Partition module. As the input images are three-channel images, each region is flattened into a one-dimensional vector in the channel direction, resulting in an image shape of 56 × 56 × 48. We specify the channel depth of the feature map as 128. The image is then linearly transformed through a Linear Embedding layer to obtain a feature map of size 56 × 56 × 128. Subsequently, the feature map goes through four different stages. We specify the number of heads for the multi-head self-attention mechanism in the four stages as 4, 8, 16, and 32, respectively.

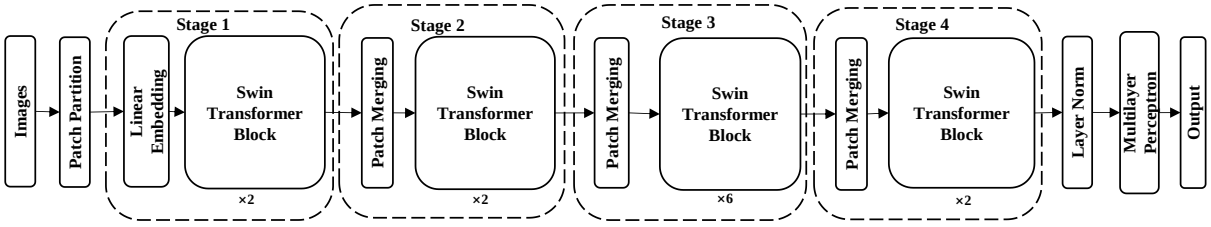


Figure 9. Structure of the Swin-transformer model.

In the experiments, the model is trained using the cross-entropy loss function, minimizing the loss to make the model predictions closer to the actual values. The Adam optimizer is utilized for optimization to enhance recognition capabilities in complex processing pathways. We set the training epochs to 25, gradually optimizing the model's performance through iterative training data updates. After each training epoch, the accuracy and loss of model are evaluated using validation data, and the best validation model during the training process is extracted, which is named by Model-a. Finally, the accuracy of the Model-a is evaluated on the test set.

2.3.2. Model-b based on KNN

Regarding the eye and eyebrow state trainer, we use the k-nearest neighbor (KNN) method. This method has a simple structure, insensitivity to outliers, and does not require assumptions about data

distribution. Initially, specific points, referred to as representative points, are selected from the training samples, to form a representative set of eye and eyebrow state. When there is a sample x to be classified, the algorithm searches for the K nearest representative points in the representative point set in the vector space. Then, the categories of these K representative points are used as candidate categories for the test sample x. Each representative point's category has a weight for the classification of the test sample, and this weight is based on the similarity between the test sample x and the K representative points. Finally, the final category of the test sample x is determined by comparing it with a predefined similarity threshold. The specific implementation steps for this method are shown in Eqs (15) and (16).

$$y(x, c_j) = \sum_{d_i \in \mathrm{KNN}} f(x, d_i, c_j, b_j) \tag{15}$$

$$f(x, d_i, c_j, b_j) = \begin{cases} 1, \sin(x, d_i)\,\delta(d_i, c_i) - b_j > 0 \\ 0, \sin(x, d_i)\,\delta(d_i, c_i) - b_j \leq 0 \end{cases} \tag{16}$$

where x represents the sample to be classified, d_i represents the i-th of the K nearest representative points, c_j represents the category $\delta(d_i, c_j) \in \{0, 1\}$, $\delta(d_i, c_j)$ takes on values in $\{0, 1\}$, where it equals 1 if d_i belongs to c_j and 0 otherwise; b_j is the threshold set for category c_j, typically determined based on the proportion of that category in the total dataset; sin (x, d_i) is the similarity between the test sample x and the representative point d_i, determined by calculating the cosine value of the angle between the corresponding vectors of the two sample points. When $y\,(x, c_j) = \max\,(y\,(x, c_i))$ (m represents the total number of categories in the sample set), the test sample x belongs to category c_j.

2.3.3. Decision-level fusion

Decision fusion refers to a method of fusing multiple decision results to improve the accuracy and stability of classification or recognition. In facial expression recognition research, multiple feature extraction methods and classifications are commonly used. To obtain different decision results, decision layer fusion can combine these different decision results to obtain more reliable and accurate recognition results.

In this paper, we address the problem of facial expression recognition in mask-obscured scenes, and integrate Swin-transformer and KNN models at the decision-making level. Specifically, we propose a recognition method that incorporates eye and eyebrow state information. By fully utilizing the state information of the eyebrows and eyes in the uncovered area of the mask, we extract facial features and estimate the facial expression. This approach maximizes the utilization of valuable information outside the mask and improves the accuracy of facial expression recognition. Specifically, as shown in Figure 10, we design two major models, Model-a and Model-b, to optimize facial expression recognition through different technical paths. Model-a is mainly based on the neural network architecture of Swin-transformer, using the Swin-transformer network to extract features from input images and train a deep learning model. On the other hand, Model-b is trained on various relative distances and angles at the eyebrows and eyes for further detection and analysis. Since Model-a and Model-b are trained and optimized in different aspects, they can provide complementary information.

In order to fully utilize the strengths of these two models, we fuse their output results at the decision level. First, Model-a predicts facial expressions on the maskeddataset. The predicted expression results are compared with Groundtruth, which is abbreviated as GT. Then, the prediction

errors of Model-a are used as input to Model-b for further detection and analysis. By analyzing eyebrow and eye state, Model-b corrects possible misclassifications in Model-a, and improves the final accuracy of facial expression recognition. This fusion method comprehensively considers the results of both models, not only increases prediction accuracy, but also provides an effective solution for facial expression recognition in complex scenarios. By combining the strengths of deep learning and traditional machine learning, our approach demonstrates excellent accuracy and robustness in the face of facial occlusion challenges.

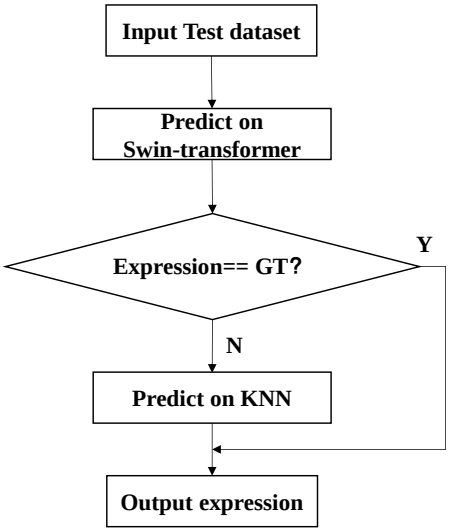


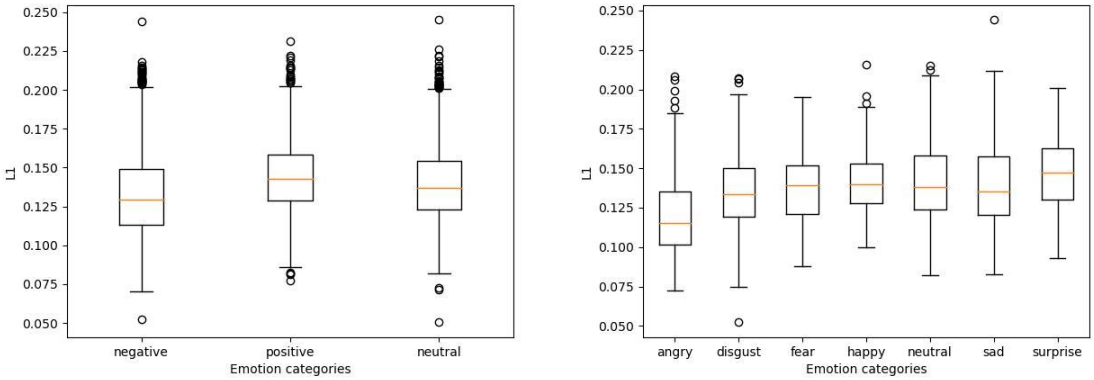
Figure 10. Decision-level fusion process diagram.

3. Experiments and results

All training frameworks are run on a PC with an 11th Gen Intel(R) Core (TM) i5 CPU, NVIDIA GeForce RTX 3050 GPU, Windows 10 operating system, and based on Python 3.9 and Pytorch 1.12.1. The input image size, batch size (patch size), and learning rate are set to 224 × 224 × 3, 8, and 0.0001, respectively. The average accuracy and precision are used as performance evaluation metrics for expression recognition, and the performance of different classification models on various expression categories is presented.

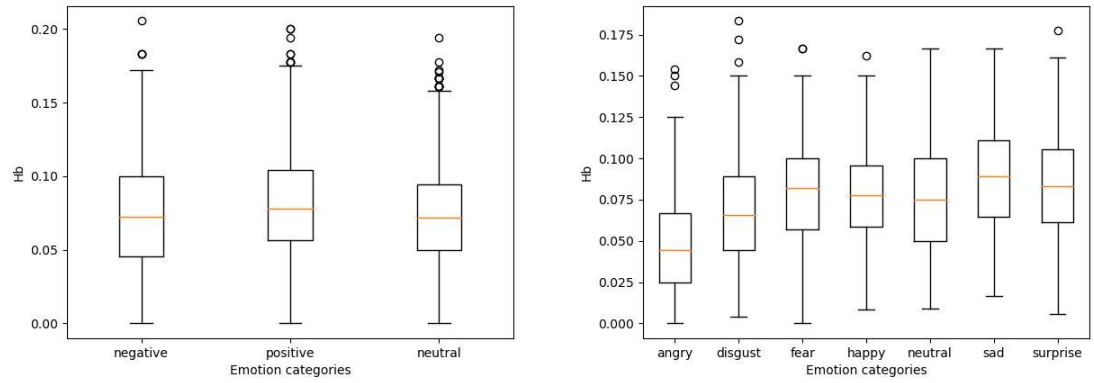
3.1. Distribution of eyebrow and eye state

According to the formula in Section 2.2, we separately analyze the distribution of eyebrow and eye state of the RAF-3 dataset and the RAF dataset for different expressions. An example of the distribution of three types of eyebrow-eye states: *L1*, *Hb*, and *He* is separately shown in Figure 11−13. In each figure, the left column presents the results for the three expressions in the RAF-3 dataset, while the right column presents the results for the seven expressions in the RAF dataset.



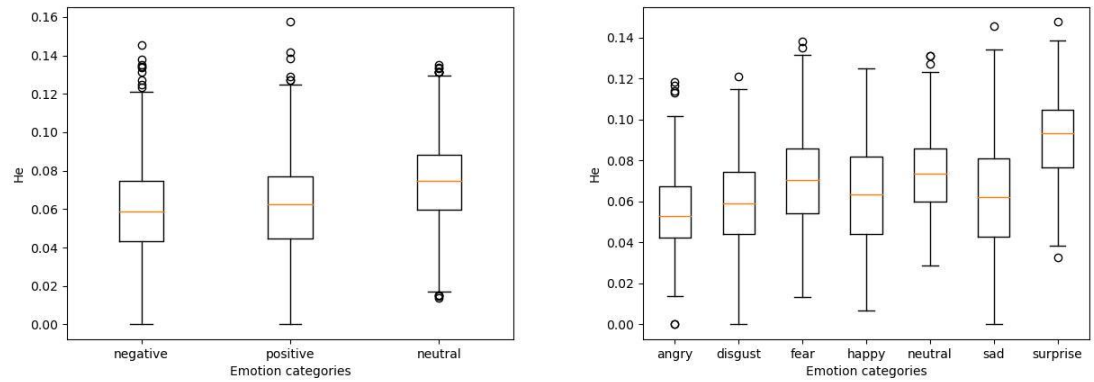
(a) (b)

Figure 11. Distribution of *L1* for different expressions. (a) RAF-3 dataset. (b) RAF dataset.



(a) (b)

Figure 12. Distribution of *Hb* for different expressions. (a) RAF-3 dataset. (b) RAF dataset.



(a) (b)

Figure 13. Distribution of *He* for different expressions. (a) RAF-3 dataset. (b) RAF dataset.

According to the box plots, it can be observed that the median of the distribution for the positive expression is the highest in terms of the *L1* and *Hb*. This indicates that when positive expressions occur, *L1* and *Hb* is generally larger. The median of negative expressions is lower, and the distribution range is relatively wide, indicating that there may be significant differences in these two types of eyebrow-eye state among individuals during negative expressions. From the seven-category box plots, it can be noted that, for *L1* and *Hb*, the distribution range for sad expressions is wide. This suggests that, when expressing sadness, there is considerable variability in the degree and frequency of these two types of eyebrow-eye state among people.

As for *He*, in terms of the three categories, the median of the neutral expression is higher than that for negative and positive expressions. It suggests that people's eyes are relatively larger in a neutral expression. The median of negative or positive expressions is low, which may be related to the tension of eye muscles during negative or positive expressions, leading to a relatively smaller eye opening. In the seven-category expressions, the median of surprise expressions is the highest, which is consistent with expectations, as surprised expressions generally involve wider eye openings.

3.2. Expression recognition results ofSwin-transformer method

Wearing a mask can affect the identification of crucial facial cues in expression recognition. Specifically, the regions available for effective expression detection are confined to the eyes, eyebrows, and forehead. To assess the performance of the Swin-transformer network under varying training and testing conditions, we conduct experiments employing datasets, such as MYRAF-3, MYRAF-3_KN95, MYLFW, and MY+LFW_KN95, as summarized in Table 3.

When trained on the MYRAF-3 Training set and tested on its corresponding Test set, the Swin-transformer achieves an average accuracy of 87.5%. However, when trained on the MYRAF-3 Training set and tested on the MYRAF-3_KN95 dataset, the accuracy decreases to 56.5%. Similarly, on the LFW-related datasets, the accuracy decreases to 43.6%. This underscores that wearing a mask has a negative impact on the average accuracy of expression recognition. Compared to training and testing on the MYRAF-3 dataset, training on the MYRAF-3_KN95 Training set and testing on the MYRAF-3 Test set, the average accuracy decreases by 19.6%, and on LFW-related datasets, the average accuracy decreases by 12.8%. This indicates that incorporating mask information during training influences the final recognition performance. Additionally, compared to training on the MYRAF-3 Training set and testing on the MYRAF-3_KN95 Test set, training and testing on the MYRAF-3_KN95 dataset leads to a 22.6% increase in the average accuracy of expression recognition, and on LFW-related datasets, the accuracy increases by 17.9%. Therefore, achieving accurate facial expression recognition with masks necessitates special treatment, and training models on datasets containing masked faces can improve the accuracy of mask expression recognition.

The results of mask expression recognition vary across different expression types. For instance, when recognizing positive expressions, the accuracy of the MYRAF-3 Training set (trained and tested) is 52.4% higher than that trained on the MYRAF-3 Training set and tested on the MYRAF-3_KN95 Test set. Under the same conditions, for LFW-related datasets, the precision is 75% higher. This indicates that wearing a mask indeed has a negative impact on the recognition of positive expressions.

Table 3. Three-class expression recognition results using Swin-transformer under different conditions.

Training data	Testing data	Expression categories	Precision(%)	Average accuracy(%)
MYRAF-3_KN95	MYRAF-3	negative	80.3	67.9
Training set	Test set	neutral	37.9	
		positive	85.4	
MYRAF-3_KN95	MYRAF-3_KN95	negative	76.6	79.1
Training set	Test set	neutral	70.4	
		positive	90.2	
MYRAF-3	MYRAF-3	negative	90.8	87.5
Training set	Test set	neutral	78.6	
		positive	93.2	
MYRAF-3	MYRAF-3_KN95	negative	87.9	56.5
Training set	Test set	neutral	40.9	
		positive	40.8	
MYLFW_KN95	MYLFW	negative	29.7	59.7
Training set	Test set	neutral	58.9	
		positive	90.5	
MYLFW_KN95	MYLFW_KN95	negative	32.4	61.5
Training set	Test set	neutral	64.6	
		positive	87.5	
MYLFW	MYLFW	negative	33.8	72.5
Training set	Test set	neutral	90.9	
		positive	92.9	
MYLFW	MYLFW_KN95	negative	14.9	43.6
Training set	Test set	neutral	98.1	
		positive	17.9	

3.3. Fusion results of different machine learning models and Swin-transformer

We conduct experiments to explore the fusion effects of different machine learning models with Swin-transformer on the MYRAF-3_KN95 dataset, including Support Vector Machine, Random Forest, Gradient Boosting Tree, as well as KNN (n_neighbors = 3), KNN (n_neighbors = 5), and KNN (n_neighbors = 7). The parameter n_neighbors represents the number of neighbors used for prediction during classification. The results are shown in Table 4. The KNN (n_neighbors = 3) model exhibits good complementarity with the Swin-transformer, demonstrating excellent performance on the MYRAF-3_KN95 dataset, withaverage precision reaching the highest levels. Based on these results, we choose the KNN machine learning model as the training model for eyebrow and eye states and integrate it with the Swin-transformer at the decision layer to achieve facial expression recognition on the masked dataset.

Table 4. Fusion results of different machine learning models and Swin-transformer on the MYRAF-3_KN95.

Machine Learning Models	Expression categories	Precision (%)	Average accuracy (%)
Support Vector Machine	negative	77.7	82.4
	neutral	70.4	
	positive	99.2	
Random Forest	negative	88.6	84.9
	neutral	71.1	
	positive	94.9	
Gradient Boosting Tree	negative	86.6	84.2
	neutral	70.4	
	positive	95.7	
KNN（n_neighbors = 5）	negative	89.0	85.4
	neutral	73.1	
	positive	94.2	
KNN（n_neighbors = 7）	negative	88.2	85.0
	neutral	72.1	
	positive	94.7	
KNN（n_neighbors = 3）	negative	**89.7**	**85.8**
	neutral	**74.1**	
	positive	93.7	

3.4. Expression recognition results of the TKNN method

3.4.1. Intra-dataset testing

To validate the effectiveness of our approach, we train Model-a on the MYRAF-3_KN95 Training set and fed the MYRAF-3_KN95 Test set into the Model-a. Among the misclassified images, there are 168 negative facial expressions, 120 neutral facial expressions, and 77 positive facial expressions. Subsequently, we input all misclassification results into Model-b trained on the MYRAF-3_KN95 Training set for further classification, with a chosen number of nearest neighbors set to 3. The results demonstrate that our method improves the average accuracy on the MYRAF-3_KN95 Test set by 6.7%. We also test our approach on the MYLFW_KN95 dataset by inputting its Test set into Model-a trained on the MYLFW_KN95 Training set. Among the misclassified images, there are 50 negative facial expressions, 113 neutral facial expressions, and 62 positive facial expressions. Subsequently, the misclassification results are input into Model-b trained on the MYLFW_KN95 Training set, and the chosen number of nearest neighbors was set to 3. The fused model shows an 8.8% improvement in average accuracy. The results of both datasets confirm the effectiveness of our method in improving the average accuracy of facial expression recognition.

In order to test the generalization ability of our method, we randomly shuffle the MYRAF-3_KN95 Training set and divided it equally into 5 parts. Each part is used as a validation set once, while the others are used as Training sets. We input these sets into the Swin-transformer for training, selecting the best model on the validation set. We then test the performance on the original Test set, repeat this process five times, calculate the mean of the results, and compare it with the results obtained

by training on the original MYRAF-3_KN95 Training set and testing on the Test set. As shown in Table 5, the average accuracy decreases by 3.1%, the accuracy of negative expressions decreases by 2.8%, the accuracy of neutral expressions decreases by 8%, and the accuracy of positive expressions increases by 1.4%. Moreover, there are no significant changes in each fold, indicating that our method does not overfit and has strong generalization ability.

Table 5. Intra-dataset testing results of the TKNN method.

Dataset	Expression categories	Precision (%)	Average accuracy (%)
MYRAF-3_KN95	negative	89.7	**85.8**
	neutral	74.1	
	positive	93.7	
MYLFW_KN95	negative	41.5	**70.3**
	neutral	74.3	
	positive	95.0	
First fold of MYRAF-3_KN95	negative	87.4	79.0
	neutral	55.5	
	positive	94.1	
Second fold of MYRAF-3_KN95	negative	87.2	85.3
	neutral	74.4	
	positive	94.2	
Third fold of MYRAF-3_KN95	negative	92.2	82.4
	neutral	63.3	
	positive	91.8	
Fourth fold of MYRAF-3_KN95	negative	85.4	83.6
	neutral	68.0	
	positive	97.4	
Fifth fold of MYRAF-3_KN95	negative	82.1	83.1
	neutral	69.5	
	positive	97.8	
Five-Fold Average of MYRAF-3_KN95	negative	86.9	**82.7**
	neutral	66.1	
	positive	95.1	

3.4.2. Cross-dataset testing

To further validate the generalization of the proposed method, we conduct cross-dataset testing on the MYLFW_KN95 and MYRAF-3_KN95 datasets, as shown in Table 6. First, input MYRAF-3_KN95 dataset to train Model-a, and utilize this model to predict MYLFW_KN95 dataset. Input the misclassifications of the prediction into the KNN machine learning model which is trained on the MYRAF-3_KN95 dataset for further classification. It can be observed that compared to Swin-transformer, our method achieves an increase of 14.5% in average accuracy, and there are improvements in the precision of all three emotion classes. Specifically, negative emotions increase by 1%, neutral emotions increase by 9.3%, and positive emotions increase by 33%. Similarly, when training on the MYLFW_KN95 dataset and testing on the MYRAF-3_KN95 dataset, the average

accuracy increases by 8.1%, and there are improvements in the precision of all three emotion classes, with negative emotions increasing by 9.5%, neutral emotions increasing by 13.8%, and positive emotions increasing by 0.9%.

Table 6. Cross-dataset testing results of Swin-transformer method and TKNN method.

Method	Training data	Testing data	Expression categories	Precision (%)	Average Accuracy (%)
Swin-transformer	MYRAF-3_KN95 Training set	MYLFW_KN95 Test set	negative	97.6	33.7
			neutral	3.2	
			positive	0.4	
Swin-transformer	MYLFW_KN95 Training set	MYRAF_KN95 Test set	negative	3.2	41.0
			neutral	32.6	
			positive	87.3	
TKNN	MYRAF-3_KN95 Training set	MYLFW_KN95 Test set	negative	98.6	**48.2**
			neutral	12.5	
			positive	33.4	
TKNN	MYLFW_KN95 Training set	MYRAF-3_KN95 Test set	negative	12.7	**49.1**
			neutral	46.4	
			positive	88.2	

3.4.3. Comparison with other methods in facial expression recognition

The performance of the proposed method and other advanced facial expression recognition methods are compared on the datasets MYRAF-3_KN95 and MYLFW_KN95, mostly including the ResNet method [40], Swin-transformer method [18], and KNN method. The results in

show that the recognition performance of TKNN method is superior to the currently leading facial expression recognition methods. In comparison to the TKNN method, the deep learning-based Swin-transformer method and ResNet method exhibit lower rankings in facial expression recognition performance across both the MYRAF-3_KN95 and MYLFW_KN95 datasets. The average accuracy on the MYRAF-3_KN95 dataset is 79.1% and 71.5% for the Swin-transformer and ResNet methods, respectively, while on the MYLFW_KN95 dataset, it is 61.5% and 50.6%, respectively. The KNN method based on instance learning achieves over 30% accuracy.

3.5. Expression recognition results with introduced eyebrow and eye feature points

At present, most deep learning-based facial expression recognition algorithms solely focus on visual information from the face, when dealing with images of individuals wearing masks. Zheng et al. [41] utilized body keypoints to enhance fall detection accuracy. We hypothesize that by incorporating feature points around the eyebrows and eyes, which are crucial regions for conveying expression information, the network may better discern subtle facial cues even in the presence of mask obstructions. We conduct experiments to explore whether marking feature points around the eyebrows and eyes could redirect the network's attention to crucial information outside the mask. We utilize the feature point detection model mentioned in Section 2.1.2 to label the feature points 17−26 and 36−47

on images with individuals wearing masks. This results in the RAF-3_KN95tzd dataset, MYRAF-3_KN95tzd dataset, LFW_FER_KN95tzd dataset, and MYLFW_FER_KN95tzd dataset. The impact of introducing feature points on the Swin-transformer and TKNN method is shown in Table 8 and Table **9**. The results indicate that introducing feature points during the training phase leads to a decrease in the average accuracy of expression recognition, whether for Swin-transformer or TKNN method.

Table 7. Results of different methods on the MYRAF-3_KN95 and MYLFW_KN95.

Method	Dataset	Expression categories	Precision (%)	Average accuracy (%)
ResNet	MYRAF-3_KN95	negative	69.3	**71.5**
		neutral	63.1	
		positive	82.2	
	MYLFW_KN95	negative	4.1	**50.6**
		neutral	63.6	
		positive	84.1	
Swin-transformer	MYRAF-3_KN95	negative	76.6	**79.1**
		neutral	70.4	
		positive	90.2	
	MYLFW_KN95	negative	32.4	**61.5**
		neutral	64.6	
		positive	87.5	
KNN	MYRAF-3_KN95	negative	53.4	**33.1**
		neutral	9.4	
		positive	36.4	
	MYLFW_KN95	negative	16.2	**35.2**
		neutral	28.8	
		positive	60.7	
TKNN	MYRAF-3_KN95	negative	89.7	**85.8**
		neutral	74.1	
		positive	93.7	
	MYLFW_KN95	negative	41.5	**70.3**
		neutral	74.3	
		positive	95.0	

Table 8. Expression recognition results of Swin-transformer before and after adding feature points around the eyebrows and eyes.

Training data	Testing data	Expression categories	Precision (%)	Average accuracy (%)
MYRAF-3_KN95	MYRAF-3_KN95	negative	76.6	**79.1**
Training set	Test set	neutral	70.4	
		positive	90.2	
MYRAF-3_KN95tzd	MYRAF-3_KN95	negative	83.1	72.9
Training set	Test set	neutral	49.3	
		positive	86.4	
MYLFW_KN95	MYLFW_KN95	negative	32.4	**61.5**
Training set	Test set	neutral	64.6	
		positive	87.5	
MYLFW_KN95tzd	MYLFW_KN95	negative	8.1	52.1
Training set	Test set	neutral	59.2	
		positive	89.1	

Table 9. Expression recognition results of TKNN before and after adding feature points around the eyebrows and eyes.

Training data	Testing data	Expression categories	Precision (%)	Average accuracy (%)
MYRAF-3_KN95	MYRAF-3_KN95	negative	89.7	**85.8**
Training set	Test set	neutral	74.1	
		positive	93.7	
MYLFW_KN95	MYLFW_KN95	negative	41.5	**70.3**
Training set	Test set	neutral	74.3	
		positive	95.0	
MYRAF-3_KN95tzd	MYRAF-3_KN95	negative	92.4	79.5
Training set	Test set	neutral	55.5	
		positive	90.5	
MYLFW_KN95tzd	MYLFW_KN95	negative	23.0	62.9
Training set	Test set	neutral	70.4	
		positive	95.4	

3.6. Expression recognition results of the seven-class expression dataset

In the RAF dataset, basic expressions are divided into seven categories. After redividing the "train" folder into a Training set and a validation set in a 9 : 1 ratio, there are 11,046 images in the Training set, 1225 images in the validation set, and 3068 images in the Test set. After adding KN95 masks to this dataset, the number of images decreases to 9838, and we name it the RAFKN95 dataset. We conduct expression recognition experiments under various training and testing conditions using Swin-

transformer, and the results are shown in Table 10. It can be observed that the average accuracy change of Swin-transformer before and after wearing masks in the seven-class expression dataset is consistent with the three-class expression recognition results in Table 3.

Table 10. Seven-class expression recognition results using Swin-transformer under different conditions.

Training data	Testing data	Expression categories	Precision (%)	Average accuracy (%)
RAF_KN95	RAF	angry	63.6	52.5
Training set	Test set	disgust	23.8	
		fear	40.5	
		happy	75.5	
		neutral	41.2	
		sad	70.3	
		surprise	52.9	
RAF_KN95	RAF_KN95	angry	55.8	58.3
Training set	Test set	disgust	31.7	
		fear	22.9	
		happy	84.1	
		neutral	82.2	
		sad	63.5	
		surprise	68.3	
RAF	RAF	angry	73.5	76.0
Training set	Test set	disgust	56.2	
		fear	59.5	
		happy	93.2	
		neutral	82.1	
		sad	82.8	
		surprise	84.5	
RAF	RAF_KN95	angry	38.5	48.3
Training set	Test set	disgust	27.7	
		fear	25.7	
		happy	52.2	
		neutral	44.9	
		sad	78.9	
		surprise	70.2	

We train Swin-transformer on the RAF_KN95 Training set and predict the misclassification results on the RAF_KN95 Test set. It is found that among the misclassified images for each expression category, there are 46 for angry, 69 for disgust, 27 for fear, 128 for happy, 75 for neutral, 111 for sad, and 65 for surprise. These misclassification results are input into the KNN model trained on the RAF_KN95 Training set, and the results are presented in Table 11. It can be observed that our method, building on Swin-transformer, has improved accuracy of 4.3%. This demonstrates the effectiveness of our method in seven-class expression recognition.

Table 11. Results of Swin-transformer and TKNN methods on the RAF_KN95.

Method	Training data	Testing data	Expression categories	Precision (%)	Average accuracy (%)
Swin-transformer	RAF_KN95 Training set	RAF_KN95 Test set	angry	55.8	58.3
			disgust	31.7	
			fear	22.9	
			happy	84.1	
			neutral	82.2	
			sad	63.5	
			surprise	68.3	
TKNN	RAF_KN95 Training set	RAF_KN95 Test set	angry	60.6	**62.6**
			disgust	38.6	
			fear	25.8	
			happy	93.7	
			neutral	82.9	
			sad	68.4	
			surprise	68.3	

4. Conclusions

The introduction of mask information can easily suppress or mask original facial expressions, affecting the average accuracy of facial expression recognition. In order to overcome this problem, we propose a facial expression recognition model, TKNN, which incorporates the state of eyebrow and eye using the Swin-transformer model and KNN model. This model can focus on the macroscopic facial features of masked images, and magnify the detailed facial features around the eyebrows and eyes, thereby improving the average accuracy of facial expression recognition. Experimental results on the MYRAF-3_KN95 and MYLFW_KN95 datasets indicate that our proposed deep learning model, which incorporates state information of eyebrow and eye, can improve the average accuracy of facial expression recognition. In the future, we plan to utilize semi-supervised or weakly supervised methods to reduce dependence on labeled data, thereby enhancing robustness to different facial expression types, mask types, and degrees of occlusion.

Use of AI tools declaration

We declare that we have not used Artificial Intelligence (AI) tools in the creation of this article.

Acknowledgments

This work was supported by the Humanities and Social Science Fund of Ministry of Education (22YJA880091), CN.

Conflict of interest

We declare that there are no conflicts of interest.

References

1. P. Ekman, Facial expression and emotion, *Am. Psychol.*, **48** (1993), 384392. https://doi.org/10.1037/0003-066X.48.4.384
2. L. Zhang, B. K. Verma, D. Tjondronegoro, V. Chandran, Facial expression analysis under partial occlusion: A survey, *ACM Comput. Surv.*, **51** (2018), 1−49. https://doi.org/10.1145/3158369
3. I. Kotsia, I. Buciu, I. Pitas, An analysis of facial expression recognition under partial facial image occlusion, *Image Vision Comput.*, **26** (2008), 10521067. https://doi.org/10.1016/j.imavis.2007.11.004
4. H. K. Wong, A. J. Estudillo, Face masks affect emotion categorisation, age estimation, recognition, and gender classification from faces, *Cognit. Res. Princ. Implic.*, **7** (2022). https://doi.org/10.1186/s41235-022-00438-x
5. H. Cooper, A. Brar, H. Beyaztas, B. J. Jennings, R. J. Bennetts, The effects of face coverings, own-ethnicity biases, and attitudes on emotion recognition, *Cogn. Res.*, **7** (2022). https://doi.org/10.1186/s41235-022-00400-x
6. F. Grundmann, K. Epstude, S. Scheibe, Face masks reduce emotion-recognition accuracy and perceived closeness, *Plos One*, **16** (2021), e0249792. https://doi.org/10.1371/journal.pone.0249792
7. M. Marini, A. Ansani, F. Paglieri, F. Caruana, M. Viola, The impact of facemasks on emotion recognition, trust attribution and re-identification, *Sci. Rep.*, **11** (2021), 5577. https://doi.org/10.1038/s41598-021-84806-5
8. L. Zhang, D. Tjondronegoro, V. Chandran, Random Gabor based templates for facial expression recognition in images with facial occlusion, *Neurocomputing*, **145** (2014), 451464. https://doi.org/10.1016/j.neucom.2014.05.008
9. P. Lucey, J. F. Cohn, T. Kanade, J. Saragih, Z. Ambada, The extended cohn-kanade dataset (ck+): A complete dataset for action unit and emotion-specified expression, in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition-Workshops*, IEEE, (2010), 94−101.
10. M. Lyons, S. Akamatsu, M. Kamachi, J. Gyoba, Coding facial expressions with Gabor wavelets, *in Proceedings of The 3rd IEEE International Conference on Automatic Face and Gesture Recognition*, *IEEE*, (1998), 200205. https://doi.org/10.1109/AFGR.1998.670949
11. H. Ding, P. Zhou, R. Chellappa, Occlusion-adaptive deep network for robust facial expression recognition, in *2020 IEEE International Joint Conference on Biometrics (IJCB)*, IEEE, (2020). https://doi.org/10.1109/IJCB48548.2020.9304923
12. E. Barsoum, C. Zhang, C. C. Ferrer, Z. Y. Zhang, Training deep networks for facial expression recognition with crowd-sourced label distribution, in *18th ACM International Conference on Multimodal Interaction*, ACM, (2016), 279283. https://doi.org/10.1145/2993148.2993165
13. A. Mollahosseini, B. Hasani, M. H. Mahoor, Affectnet: A database for facial expression, valence, and arousal computing in the wild, *IEEE Trans. Affective Comput.*, **10** (2017), 18–31. https://doi.org/10.1109/TAFFC.2017.2740923

14. K. Wang, X. J. Peng, J. F. Yang, D. Meng, Y. Qiao, Region attention networks for pose and occlusion robust facial expression recognition, *IEEE Trans. Image Process.*, **29** (2020), 4057–4069. https://doi.org/10.1109/TIP.2019.2956143
15. S. Li, W. Deng, J. P. Du, Reliable crowdsourcing and deep locality-preserving learning for expression recognition in the wild, in *Proceedings of The IEEE Conference on Computer Vision and Pattern Recognition*, *IEEE*, (2017), 2852–2861.
16. A. Dapogny, K. Bailly, S. Dubuisson, Confidence-weighted local expression predictions for occlusion handling in expression recognition and action unit detection, *Int. J. Comput. Vision*, **126** (2018), 255–271. https://doi.org/10.1007/s11263-017-1010-1
17. H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, H. Jegou, Training data-efficient image transformers & distillation through attention, in *International Conference on Machine Learning*, (2021), 10347–10357
18. Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, et al., Swin transformer: Hierarchical vision transformer using shifted windows, in *Proceedings of The IEEE/CVF International Conference on Computer Vision*, *IEEE*, (2021), 10012–10022. https://doi.org/10.1109/ICCV48922.2021.00986
19. D. Poux, B. Allaert, N. Ihaddadene, I. M. Bilasco, C. Djeraba, M. Bennamoun, et al., Dynamic facial expression recognition under partial occlusion with optical flow reconstruction, *IEEE Trans. Image Process.*, **31** (2021), 446–457. https://doi.org/10.1109/TIP.2021.3129120
20. J. Lou, Y. Wang, C. Nduka, M. Hamedi, I. Mavridou, F. Y. Wang, Realistic facial expression reconstruction for VR HMD users, *IEEE Trans. Multimedia*, **22** (2019), 730–743. https://doi.org/10.1109/TMM.2019.2933338
21. L. Itti, C. Koch, Computational modelling of visual attention, *Nat. Rev. Neurosci.*, **2** (2001), 194–203. https://doi.org/10.1038/35058500
22. Y. Li, J. Zeng, S. Shan, X. Chen, Occlusion aware facial expression recognition using CNN with attention mechanism. *IEEE Trans. Image Process.*, **28** (2018), 2439–2450.
23. S. Liu, W. Guo, Y. Zhang, X. Cheng, Robust regularized encoding for random occlusion facial expression recognition, *CAAI Trans. Intell. Syst.*, **13** (2018), 261–268. https://doi.org/10.11992/tis.201609002
24. X. Ben, M. Yang, P. Zhang, J. Li, Overview of automatic micro-expression recognition, *J. Comput. Aided Design Comput. Graphics*, **26** (2014), 1385–1395.
25. S. Ramachandra, S. Ramachandran, Region specific and subimage based neighbour gradient feature extraction for robust periocular recognition, *J. King Saud. Univ. Comput. Inf. Sci.*, **34** (2022), 7961–7973. https://doi.org/10.1016/j.jksuci.2022.07.013
26. M. Okawa, Synergy of foreground-background images for feature extraction: Offline signature verification using Fisher vector with fused KAZE features, *Pattern Recognit.*, **79** (2018), 480–489. https://doi.org/10.1016/j.patcog.2018.02.027
27. N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, in *Proceedings of The 2005 IEEE Computer Society Conference on Computer Vision And Pattern Recognition*, (2005), 886–893. https://doi.org/10.1109/CVPR.2005.177f
28. B. Huang, Z. Wang, G. Wang, Z. Han, K. Jiang, Local eyebrow feature attention network for masked face recognition, *ACM Trans. Multimedia Comput. Commun. Appl.*, **19** (2023). https://doi.org/10.1145/3569943

29. K. Zheng, K. Ci, H. Li, L. Shao, G. Sun, J. Liu, et al., Heart rate prediction from facial video with masks using eye location and corrected by convolutional neural networks, *Biomed. Signal Process. Control*, **75** (2022), 103609. https://doi.org/10.1016/j.bspc.2022.103609
30. P. Viola, M. J. Jones, Robust real-time face detection, *Int. J. Comput. Vision*, **57** (2004), 137–154. https://10.1023/B:VISI.0000013087.49260.fb
31. D. Li, Y. Ren, T. Du, W. Liu, Eyebrow semantic description via clustering based on Axiomatic Fuzzy Set, *Wiley Int. Rev. Data Mining Knowl. Discovery*, **8** (2018), e1275. https://doi.org/10.1002/widm.1275
32. J. Zhang, K. Zheng, S. Mazhar, X. Fu, J. Kong, Trusted emotion recognition based on multiple signals captured from video, *Expert Syst. Appl.*, **233** (2023), 120948. https://doi.org/10.1016/j.eswa.2023.120948
33. H. Tao, Q. Duan, M. Lu, Z. Hu, Learning discriminative feature representation with pixel-level supervision for forest smoke recognition, *Pattern Recognit.*, **143** (2023), 109761. https://doi.org/10.1016/j.patcog.2023.109761
34. H. Tao, Q. Duan, Hierarchical attention network with progressive feature fusion for facial expression recognition, *Neural Networks*, **170** (2024), 337–348.
35. A. Anwar, A. Raychowdhury, Masked face recognition for secure authentication, preprint, arXiv: 2008.11104. https://arXiv.org/2008.11104v1.
36. V. Kazemi, J. Sullivan, One millisecond face alignment with an ensemble of regression trees, in *Proceedings of The IEEE Conference on Computer Vision and Pattern Recognition*, (2014), 1867–1874. https://doi.org/10.1109/CVPR.2014.241
37. B. Yang, J. Wu, G. Hattori, Facial expression recognition with the advent of face masks, in *Proceedings of The 19th International Conference on Mobile And Ubiquitous Multimedia*, (2020), 335–337. https://doi.org/10.1145/3428361.3432075
38. B. Huang, M. Mattar, T. Berg, E. Learned-Miller, Labeled faces in the wild: A database for studying face recognition in unconstrained environments, in *Workshop on Faces In 'Real-Life' Images: Detection, Alignment, And Recognition,* (2008).
39. K. Zhang, Z. Zhang, Z. Li, Y. Qiao, Joint face detection and alignment using multitask cascaded convolutional networks, *IEEE Signal Process. Lett.*, **23** (2016), 1499–1503. https://doi.org/10.1109/LSP.2016.2603342
40. K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in *Proceedings of The IEEE Conference on Computer Vision and Pattern Recognition,* (2016), 770–778. https://doi.org/10.1109/CVPR.2016.90
41. K. Zheng, B. Li, Y. Li, P. Chang, G. Sun, H. Li, et al., Fall detection based on dynamic key points incorporating preposed attention, *Math. Biosci. Eng.*, **20** (2023), 11238–11259. https://doi.org/10.3934/mbe.2023498

AIMS Press