Neural Processing Letters (2023) 55:11109–11130
https://doi.org/10.1007/s11063-023-11367-1

Swin-Fusion: Swin-Transformer with Feature Fusion for Human Action Recognition

Tiansheng Chen[1] · **Lingfei Mo**[1]

Accepted: 10 July 2023 / Published online: 20 July 2023

Abstract
Human action recognition based on still images is one of the most challenging computer vision tasks. In the past decade, convolutional neural networks (CNNs) have developed rapidly and achieved good performance in human action recognition tasks based on still images. Due to the absence of the remote perception ability of CNNs, it is challenging to have a global structural understanding of human behavior and the overall relationship between the behavior and the environment. Recently, transformer-based models have been making a splash in computer vision, even reaching SOTA in several vision tasks. We explore the transformer's capability in human action recognition based on still images and add a simple but effective feature fusion module based on the Swin-Transformer model. More specifically, we propose a new transformer-based model for behavioral feature extraction that uses a pre-trained Swin-Transformer as the backbone network. Swin-Transformer's distinctive hierarchical structure, combined with the feature fusion module, is used to extract and fuse multi-scale behavioral information. Extensive experiments were conducted on five still image-based human action recognition datasets, including the Li's action dataset, the Stanford-40 dataset, the PPMI-24 dataset, the AUC-V1 dataset, and the AUC-V2 dataset. Results indicate that our proposed Swin-Fusion model achieves better behavior recognition than previously improved CNN-based models by sharing and reusing feature maps of different scales at multiple stages, without modifying the original backbone training method and with only increasing training resources by 1.6%. The code and models will be available at https://github.com/cts4444/Swin-Fusion.

Keywords Action recognition · Swin-Transformer · Feature pyramid · Image classification

✉ Lingfei Mo
lfmo@seu.edu.cn

Tiansheng Chen
tschen@seu.edu.cn

[1] School of Instrument Science and Engineering, Southeast University, Nanjing 210096, Jiangsu, China

1 Introduction

The automatic recognition of various human activities in varied situations is known as human action recognition (HAR). The HAR model infers human actions such as drinking, jogging, and riding from information in the image or video, such as objects, backdrop, and human posture. HAR has been one of computer vision's most critical study issues in recent years. Understanding human actions in computer vision has been the subject of much research. HAR has tremendous application values in Human-Computer Interaction (HCI), image annotation, recognition of specific behaviors, video retrieval, and video surveillance.

Sensor-based HAR and vision-based HAR are two HAR approaches that utilize distinct data types. Meanwhile, sensor-based HAR can be categorized as external sensor-based HAR and wearable sensor-based HAR. The devices are fixed in the preset points of interest in the former. The discrimination of the action depends entirely on the range of the user-sensor interaction, but this exposes the target individual to a high number of monitors to precisely recognize the human action. These restrictions do not apply to the latter wearable sensors, which can track a variety of data about the target person, including motion data (acceleration, position), environmental data (temperature, humidity), and physiological data (heart rate, blood pressure). Since sensor-based HAR requires the purchase of additional sensor devices and has to consider issues such as wearing comfort and different application scenarios, a large amount of HAR research in recent years has focused on more convenient vision-based HAR. Vision-based HAR can be further subdivided into video-based HAR and still image-based HAR. Notably, the video-based HAR has been extensively investigated in recent years and has yielded many excellent outcomes, including TSM [1], UniFormer [2], and Omnivore [3]. The video consists of many frames, each of which is a still image. The frames contain information about the object's motion, allowing the full use of both temporal and spatial dimensions for more effective behavior recognition than still image-based approaches. The still image-based HAR is challenging to recognize motion trajectories and lacks temporal cues. It cannot use Spatio-temporal features to characterize actions. In addition, unfavorable factors such as cluttered backgrounds and postural occlusion in images make action recognition more challenging. However, the action recognition model must fit the data well and have good generalization ability. At the same time, a limited number of datasets are currently available for action recognition from still images. Hence, action recognition based on still images is more complex and challenging. However, it is possible to achieve this, as people can often recognize the type of human action from a still image. Video is not essential. With the rapid growth of the Internet, it is now commonplace for people to take photos on their cell phones and upload them to the web[4]. With an increasing number of photos of human actions being uploaded to the web and preserved in enormous image databases, behavior identification based on still images has received much interest in recent years. Developing an efficient behavior recognition model is crucial to comprehend still images.

Although there have been many practical advancements in still image-based action identification over the past decade, the majority of the approaches [5–8] are based on improved CNNs. Researchers have advanced the action recognition field by continuously improving classical CNN models such as ResNet and Inception. However, it is difficult to have a holistic understanding of human behavior in a global structure with limited information from a still image due to the fixed receiver field of CNNs' lack of long-range perception capability [9]. During deep CNNs training, the overfitting issue typically arises because of the need for a substantial amount of labeled still image data. In contrast, the transformer can calculate the similarity and relationship between various pixels using the attention mechanism, giving

more weight to regions it wishes to focus on and less to irrelevant parts. It is also capable of determining the relationship between distant pixels, which is essential for behavior identification tasks and can aid in locating the location between significant behavior points. Therefore, transformer structures are better appropriate for behavior recognition tasks than CNNs. Similarly, we discovered that still image-based HAR using transformer-based models such as VIT and Swin-Transformer had received less attention. To advance the development of pure transformer-based models in this field, we use the Swin-Transformer, which includes the long-range feature, to study HAR based on still images, which can effectively mitigate the issues caused by the above CNN.

In this study, we develop the Swin-Fusion, a new still image-based HAR model based on the Swin-Transformer. Due to the self-attentive mechanism of the Swin-Transformer, the model has a global receptive field, and the multi-head attention mechanism guarantees that the network can focus on multiple discriminative features[10, 11]. The shifted window-based self-attention of the Swin-Transformer enables the self-attention computation to be limited to a non-overlapping, fixed-size zone and also permits cross-window connections, which enhances performance with a far lower computational cost than VIT. In the case of giving full play to the advantages of the original model, based on Swin-Transformer, we use the unique hierarchical structure of the model itself, skillfully combine Feature Pyramid Networks (FPN), which is widely used in target detection and segmentation, and perform Modified to make it general for static image based HAR tasks. The new feature fusion module does not change the training method of the original backbone. It only adds a small number of training resources to share and reuse the features of each stage, effectively improving the behavior recognition effect.

To demonstrate the superiority of our model, we ran exhaustive tests on five HAR datasets based on still images. The final experimental results demonstrate that our model outperforms advanced still image-based HAR techniques. Our primary contributions are summarized here:

- Swin-Fusion is a novel vision-based HAR classification model that can considerably improve the accuracy of still image-based HAR datasets. According to our survey results, Swin-Fusion is the first still image-based HAR technique that uses Swin-Transformer as its backbone network and simultaneously fuses multi-scale behavioral features.
- We validate the effectiveness of Swin-Fusion in still image-based HAR tasks, overcoming the lack of global receptive field associated with CNNs.In particular, it has reduced computing costs and pays more attention to local attention than the conventional VIT, and the new feature fusion module delivers better behavior recognition outcomes. It is worth noting that FPN, widely used in target detection and segmentation, can be applied to still image-based HAR classification tasks with modifications after being modified to improve the final results.
- We used transfer learning and fine-tuned the ImageNet-22k pre-trained network weights to conduct extensive experiments on the Li's action dataset, the PPMI-24 dataset, the Stanford-40 dataset, the AUC-V1 dataset, and the AUC-V2 dataset. Swin-Fusion can increase the accuracy of both coarse-grained and fine-grained datasets for behavior recognition by conducting a simple but effective feature fusion on the output of the final three stages in the model.

2 Related Work

2.1 Action Recognition in Still Images

Early on, HAR extensively used conventional machine learning techniques, including Bayesian networks, support vector machines, and random forests. In order to provide generalized target categorization of images, Csurka et al. [12] extracted bags of keypoints as feature vectors and compared the outcomes on two classifiers separately. They discovered that the action recognition outcomes on support vector machines were superior to those on naive Bayes. Afterward, [13] used Linear Discriminant Analysis (LDA) to distinguish better feature vectors, combined with a rectangle histogram approach to differentiate actions, and finally, binary SVM for classification. However, the model could not work well in the case of complex backgrounds. Yao et al. [14] used a random forest with discriminative decision trees for action recognition. They performed classification at each tree node, searched for valuable and differentiated image regions by training the node and its upstream nodes, concatenated the obtained histograms, and took the intersection to form a feature representation.

Traditional machine learning algorithms perform well with small training sample sizes but require extensive pre-processing and hand-engineering to extract useful features and cannot handle the task of tens of thousands or millions of picture datasets in the present day. Then in 2012, AlexNet [15] emerged, which used convolutional neural networks (CNNs) to categorize pictures and won first place in the ILSVRC-2012 competition by a large margin, igniting a boom in the application of neural networks. Since then, numerous classical CNNs, such as VGGNet [6], ResNet [16], and GoogLeNet [17], have been proposed. Researchers have merged standard machine learning methods with CNNs for HAR to enhance the accuracy of action recognition datasets significantly. Lavinia [6] et al. combined three CNN models, supplied the concatenated data to random forests and support vector machines for classification, and fine-tuned action identification better than other advanced methods. Sreela et al. [18]employed a pre-trained residual neural network (ResNet) to extract features from still images, then used SVM to categorize the collected features, and assessed the findings on the PASCAL VOC2012 Action Dataset and the Stanford-40 Dataset with good results. Qi et al. [5] used a joint learning approach to integrate encoded pose cues into CNNs. In addition, several works [19, 20] employ selected search boxes to generate object proposals for action recognition.

In recent years, the Transformer model in NLP has been used to image classification tasks and achieved better results than CNNs. Consequently, many CNNs are being integrated with the Transformer model's attention mechanism and applied to still image-based HAR tasks. By computing similarity and weights, the attention mechanisms enable the model to focus more on the higher-weight context. Mohammadi et al. [8] used transfer learning and added the attention mechanisms to CNN to extract useful features to improve the model's action classification ability. et al. Hirooka et al. [7] also utilized the transfer learning approach, connecting features generated by four CNN branches, in conjunction with multichannel attention, to extract more useful contextual information in the feature map, achieving better results than [8], with an accuracy of 93.76% on the Stanford-40 dataset. Due to the complexity of still image-based HAR tasks, there currently needs to be more research work compared to video-based HAR. Although the CNN mentioned above approaches or CNN methods paired with attention mechanisms have achieved better results and increased the accuracy of HAR datasets based on still images, they limit the capacity of models to capture and utilize long-term pixel dependencies.

The long-range feature of our Swin-Fusion model can effectively solve the issues above, whether in the shallow or deep feature map, by utilizing adequate global information to extract behavioral features and having an overall understanding of the various behavioral details of the human body.

2.2 Swin-Transformer

Vision Transformer (VIT) [21] combines Computer Vision (CV) and Natural Language Processing (NLP) domain expertise, allowing researchers to achieve or surpass SOTA performance on a variety of tasks without using CNN structures. Unlike previous convolutional methods, VIT is a pure transformer model, which flattens the split image into a sequence and feeds it to the encoder part of the transformer. VIT will finally classify the images through fully connected layers. VIT can achieve superior results with fewer training resources than SOTA's CNN. However, the computational overhead of VIT is still high. VIT requires pre-training on a sufficiently large dataset (ImageNet-21k or JFT-300M) and migration to a task with fewer data to achieve excellent results. Numerous subsequent works [22–24] have enhanced VIT and produced SOTA results in various vision tasks.

Swin-Transformer, a recently developed transformer by Liu et al. [25] for vision tasks, won the best paper award at ICCV 2021 and demonstrated superior performance to traditional VIT in image classification, target detection, and segmentation. Swin-Transformer, unlike VIT, is a hierarchical structure similar to CNN in which the number of patches decreases rather than remains constant as the network depth increases. Furthermore, Swin-Transformer's shifted window-based self-attention restricts the attention computation to the window while maintaining the connection between windows, thereby reducing the computation overhead compared to VIT. Our model uses Swin-Transformer as the network backbone to extract behavioral characteristics.

2.3 Multi-stage Feature Extraction and Fusion

The original image pyramids were derived from traditional Gaussian pyramid structures such as Scale Invariant Feature Transform (SIFT) [26] and later Histogram of Oriented Gradient (HOG) [27]. Gaussian pyramids can get richer feature information than a single scale by mimicking the property that when a person looks at something, the scale size of the object viewed will vary according to the viewing distance. The feature information obtained will also vary. However, if this Gaussian pyramid is directly applied to CNN, it demands significant computation and memory. As a result, much research has been devoted to extracting and fusing features using different CNN layers. Long et al. [28] created a more accurate segmentation result by combining deep semantic information with shallow appearance information. Hariharan et al.[29] flattened and concatenated the different levels of feature maps generated by pyramid pooling and fed them into a fully connected layer for classification. Ghiasi et al. [30] used the multi-level laplacian pyramid reconstruction structure to efficiently integrate the underlying location information with the higher-level semantic information. Later, Lin et al. [31] proposed a new Feature Pyramid Network (FPN) that utilizes feature maps at different scales in CNNs as different levels in the feature pyramid, fusing multiple layers of feature information. FPN can effectively handle the problem of multi-scale variation during object detection and obtain more robust semantic information, striking a good balance between precision and speed. They added FPN to the fundamental Faster R-CNN algorithm and used the COCO dataset, outperforming all single-model detection methods at the time.

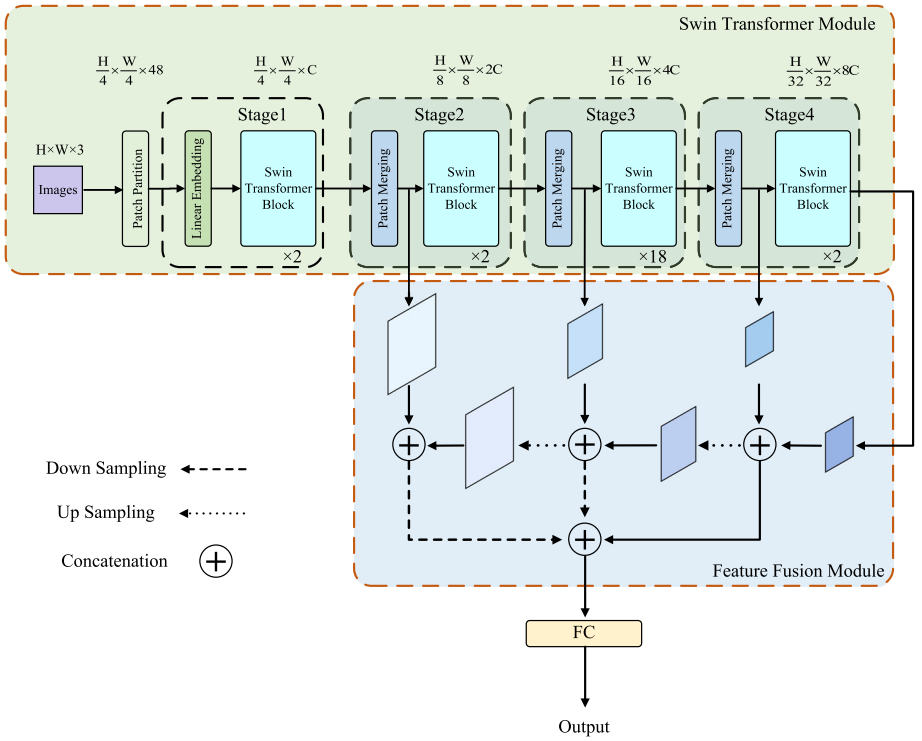

Fig. 1 The architecture of Swin-Fusion

According to previous studies, it is possible to extract and fuse the features of each layer in a hierarchical structure like CNNs. After that, we attempted to add the above conventional FPN structure to the hierarchical Swin-Transformer, but the result could have been better. Later, following our enhancement, the enhanced FPN is introduced to the Swin-Transformer feature fusion module. Through extensive experiments and continual modifications, the feature fusion module was made generalizable to different still image-based HAR datasets, further demonstrating the effectiveness of our feature fusion module in classification tasks.

3 Method

We propose a Swin-Fusion model that integrates the Swin-Transformer and Feature Pyramid architectures. Figure 1 depicts the model architecture, and we use Swin-Large (Swin-L) as the backbone network. The Swin-Transformer module can extract crucial information from images and remote dependencies between human keypoints. During the training, the $H \times W \times 3$ RGB images inputted into the model are partitioned into patches and undergo four stages. Throughout this process, each stage generates feature maps at different scales. We employ a feature fusion module to extract and merge multi-scale behavioral features, taking the output of each Patch Merging in the final three stages and the output of Stage 4 as inputs for the feature fusion module. The feature fusion module enhances the ultimate recognition accuracy by employing simple yet effective upsampling, downsampling, and concatenation methods

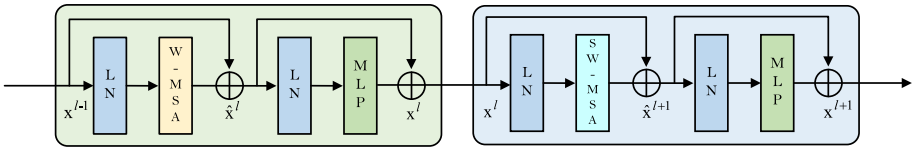

Fig. 2 Two successive Swin-transformer blocks

on the four inputs from the Swin-Transformer Module. Finally, the output is obtained through a Layer Norm layer, global pooling layer, and fully connected layer to produce the ultimate output.

3.1 Swin-Transformer Module

The Swin-Transformer module in the upper half of Fig. 1 shows the structure of the Swin-Large. First, the RGB image of H×W×3 is input into the Patch Partition module and divided into non-overlapping patch sets, one patch for every 4×4 adjacent pixels, and then flattened in the direction of the channel. Since each patch has 4×4 = 16 pixels and each pixel has three values for R, G, and B, after flattening, each patch's feature dimension is 48. The image's shape changes from [H, W, 3] to [H/4, W/4, 48] following Patch Partition. Then, the Linear Embedding layer in stage 1 linearly transforms the channel data of each pixel, and the number of channels is changed from 48 to C. The image's shape is changed from [H/4, W/4, 48] to [H/4, W/4, C]. After that, Patch Merging in stage 2 can combine each 2×2 patches, then the shape of the image in stage 2 becomes [H/8, W/8, 2C]. Patch Merging in stages 3–4 also accomplishes the same. Each Patch Merging operation will divide the H and W of the output feature map in the last stage by two and multiply C by 2. A detailed description of the Patch Merging process is presented in Sect. 3.4.

Except for Stage 1, which passes through a Linear Embedding layer, the following three stages pass through a Patch Merging layer and then repeatedly stack several Swin-Transformer blocks. Depending on the version of the Swin-Transformer model, the number of blocks varies. We employ Swin-L, and the number of blocks in stages 1–4 is [2,2,18,2]. As illustrated in Fig. 2, there are two successive blocks here. Because these two blocks are utilized in pairs, a W-MSA block is used first, followed by an SW-MSA block.

The number of blocks stacked here is even. Each MSA, SW-MSA, and MLP is preceded by a LayerNorm (LN) layer, and each module is connected through a residual connection. The following are the equations for each portion of the figure:

$$\hat{x}^{l} = W\text{-}MSA(LN(x^{l-1})) + x^{l-1} \tag{1}$$

$$x^{l} = MLP(LN(\hat{x}^{l})) + \hat{x}^{l} \tag{2}$$

$$\hat{x}^{l+1} = SW\text{-}MSA(LN(x^{l})) + x^{l} \tag{3}$$

$$x^{l+1} = MLP(LN(\hat{x}^{l+1})) + \hat{x}^{l+1} \tag{4}$$

$\hat{x}^{l}$ denotes the output features of the W-MSA module, and $\hat{x}^{l+1}$ denotes the output features of the SW-MSA module. x^{l} and x^{l+1} denote the output features of the MLP module. Section 3.3 presents a comprehensive introduction to the W-MSA and SW-MSA.

The Swin-Transformer module generates feature maps of varying sizes through four Stages, creating a pyramid-shaped feature set. Finally, the outputs of Patch Merging in the last three stages and the final output of stage 4 are inputs to the feature fusion module.

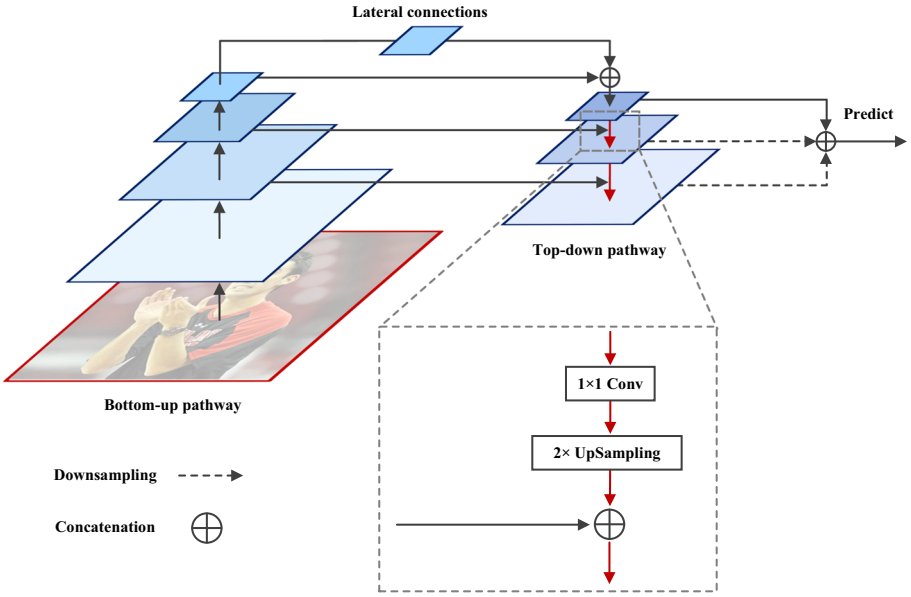

Fig. 3 The overall process of feature fusion module

3.2 Feature Fusion Module

The feature fusion module presented in this research is founded on simplicity and efficacy, allowing it to achieve maximum improvement in numerous still image-based action detection datasets with only a modest increase in training resources. We have built six distinct feature fusion modules for this purpose, and Experiment 4.4.1 compares their structure and performance. The module with the most significant enhancement effect was chosen as the feature fusion module for Swin-Fusion. To prevent Swin-Fusion from overfitting to a specific scale and decreasing the model's capacity to generalize, we utilize a lateral connection and top-down strategy to combine feature maps of multiple resolutions and channel counts. As shown in Fig. 3, the bottom-up pathway on the far left of the figure represents the forward propagation portion of Swin-Fusion. Each stage, beginning with the bottom input image, can generate feature maps of varying scales for four feature maps. In the lower stages, high image resolution and few feature channels enable learning fine-grained behavioral information. Low spatial resolution and many feature channels are more useful for modeling coarse-grained behavioral information in the higher stages.

The features generated by the three preceding stages are connected laterally to the Feature Fusion module. It should be noticed that, among the four lateral connections, the top connection is the stage 4 feature map after the transformer block. However, the remaining three connections are the stage 2-4 feature maps after Patch Merging but not after Transformer Blocks. In the top-down pathway on the right side of the figure, two feature maps of the same size are concatenated at a time. Contrary to the black down arrow, the red down arrow indicates that the number of channels in the previous layer's feature map will be changed by 1×1 convolution. In order to concatenate the feature map horizontally from the left, it is necessary to apply bilinear upsampling twice while maintaining the number of channels. As a result, a top-down feature pyramid structure is built. Then, the feature map obtained via

the top-down pathway is merged, the feature maps of the bottom two layers are upsampled to the size of the top feature map, and the three feature maps of equal size are concatenated.

The aforementioned process can be mathematically represented as follows. For conciseness, the feature pyramids constructed starting from stage 2 are sequentially referred to as ffm^0, ffm^1, and ffm^2 (ffm^1 and ffm^2 have the same size). In the feature pyramid of the Swin-Transformer module, the multi-scale features are denoted as P_1, P_2, P_3, and P_4, arranged in descending order of resolution (with downsampling multiple of 2^3, 2^4, 2^5, and 2^5 relative to the input image, respectively). Within the feature fusion module, the multi-scale features, namely K_1, K_2, and K_3, are organized in descending order of resolution. In ffm^0, the low-level features (high resolution) are located in the shallow layers of the backbone network, possessing smaller receptive fields and less semantic information. In contrast, the high-level features (low resolution) are found in the deep layers of the backbone network, exhibiting larger receptive fields and richer semantic information. However, while high-level features capture global context and semantic information, they need more detailed local information. To address this, the FFM network utilizes the high-level features from ffm^0 for upsampling to supplement semantic information. In contrast, the low-level features are provided through lateral connections to preserve local details. The network structure can be mathematically expressed as follows:

$$K_i^l = f(K_{i+1}^{l+1}, P_i^l) \tag{5}$$

Here, i (1,2,3)represents the level of the feature. l (0,1,2) represents the level of the feature pyramid. f denotes the method for fusing multi-scale features. After scaling them to the same size, the FFM network adopts the concatenation of features. The mathematical description above outlines the fusion approach, while the fusion method represented by f includes upsampling, downsampling, and concatenation, as explained in Sect. 4.4.1. The above is the whole process of the feature fusion module.

Through the lateral connection and top-down strategy described above, the model can share and reuse different-size feature maps of different stages and combine the features of each layer to accomplish more sophisticated behavior recognition. It maximizes the accuracy of behavior identification without changing the original backbone training method and with only a little increase in training resources.

3.3 W-MSA and SW-MSA

In VIT, the model directly performs Multi-head Self-Attention (MSA) on the global feature map. However, Swin-Transformer divides the original feature map into multiple disjoint regions. Multi-head Self-Attention is performed only within their respective windows by proposing the concept of Windows Multi-head Self-Attention (W-MSA). W-MSA can save much computation compared to VIT's MSA, especially when the shallow feature map is large. In most vision tasks, distinct parts of the same object or different objects with similar semantics are adjacent in the image. Self-attention in a limited window after segmentation is logical and global computation in VIT would be computationally wasteful.

The left side of Fig. 4 is the MSA module in VIT. Each patch in the feature map needs to do the self-attention calculation with all the patches. The W-MSA module on the right side of the figure first divides the feature map into four small windows according to the size of M×M (M=2 in this case) and then performs self-attention on the inside of each small window individually. The formulas for computing MSA and W-MSA are provided below:

$$\Omega(MSA) = 4hwC^2 + 2(hw)^2C$$

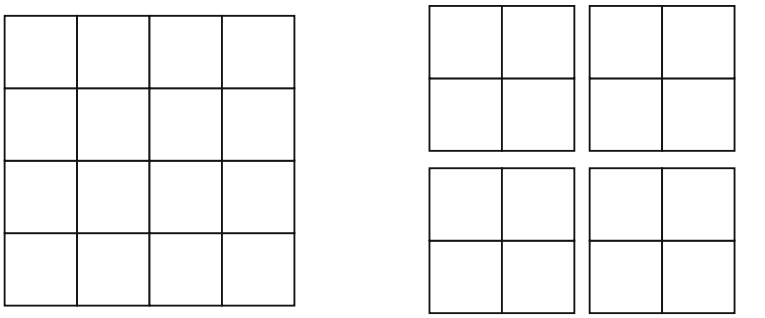

Fig. 4 MSA and W-MSA

$$\Omega(W-MSA) = (h/M)(w/M)(4M^2C^2 + 2M^4C) \tag{6}$$

$$= 4hwC^2 + 2hwM^2C \tag{7}$$

Where h is the height of the feature map, w is the width, C is the depth of the feature map, and M is the size of each window. After obtaining (h/M)×(w/M) windows for W-MSA, the MSA module will be utilized in each window. The number of the window patches is substantially lower than the number of image patches, and the image size has a linear relationship with the computation complexity of W-MSA. In comparison, VIT's MSA does a self-attention calculation on the entire map, and the computation is significantly greater than Swin-Transformer. Assuming the h and w of the feature map are 112, M = 7, and C = 128, the W-MSA module saves 40,124,743,680 FLOPs compared to the MSA module.

However, the W-MSA mentioned above has the disadvantage of isolating information transfer across separate windows. Hence, Liu et al. introduced Shifted Windows Multi-Head Self-Attention (SW-MSA), which facilitates information transfer between nearby windows via shifted windows. As shown in Fig. 5, the left side employs W-MSA (assumed to be layer L) by relocating the W-MSA windows to the right and below by a distance of $\lfloor M/2 \rfloor$ patches, as indicated in b. The 2×4 window at position two on the L+1st floor in c permits the exchange of information between windows 1 and 2 on the Lth floor. The 4×4 window located at position five on the L+1st level permits the transmission of information with four windows 1 through 4 on the Lth floor. SW-MSA solves the problem that information cannot be transmitted between various windows.

When SW-MSA is adjusted from four to nine windows, the model introduces an efficient batch computation for shifting configuration to solve the increment in computation caused by the increasing number of windows. By rearranging the windows in Fig. 6, number 5 becomes a separate window, 6 and 4 become a single window, 8 and 2 become a single window.9, 7, 3, and 1 become a single window. Currently, there are four 4×4 windows, and the amount of calculation is the same as it was previously.

3.4 Patch Merging

A Patch Merging layer first downsamples the remaining three stages, except for stage 1. Patch Merging operates similarly to pooling, but instead of taking the maximum or average values within small windows, Patch Merging extracts the values at the same positions within each small window, concatenates them to form new patches, and then concatenates all the patches

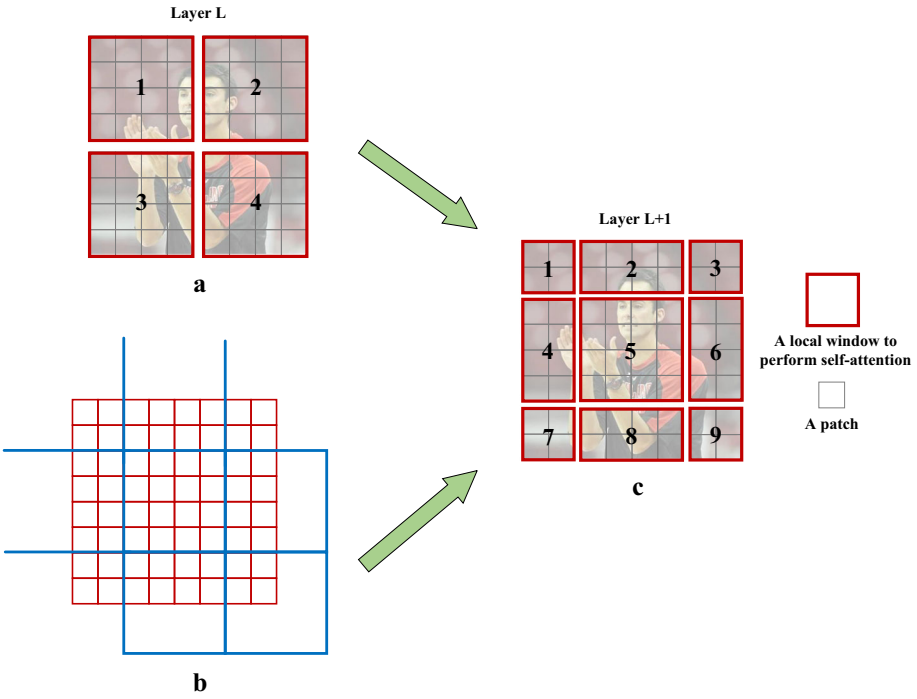

Fig. 5 The process from W-MSA to SW-MSA

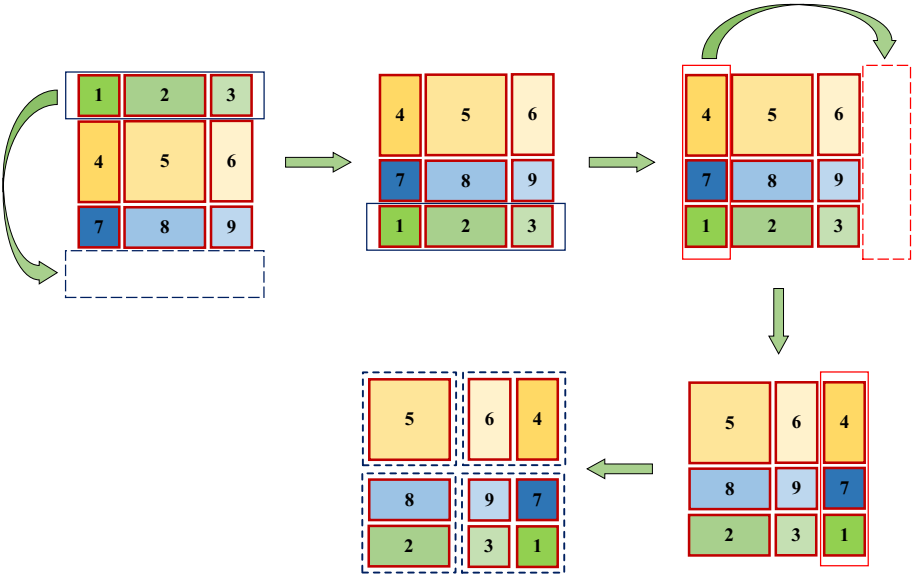

Fig. 6 Efficient batch computation for shifted configuration

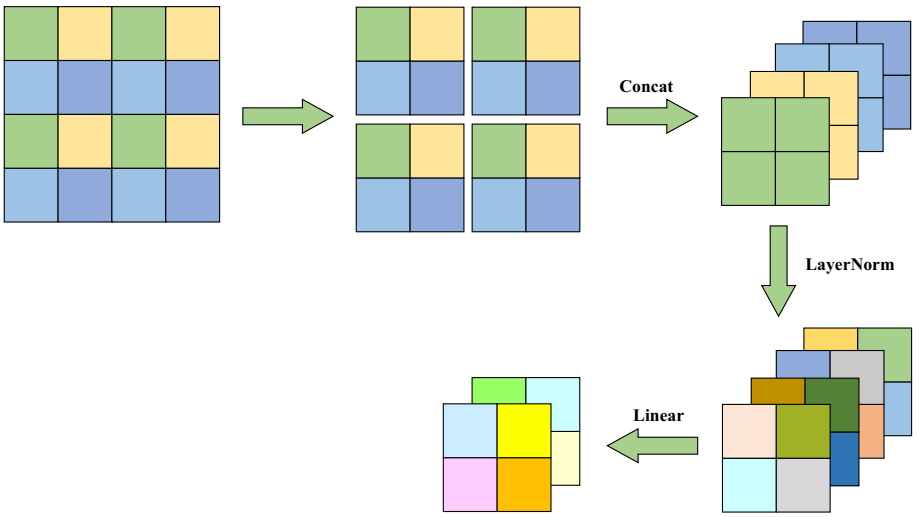

Fig. 7 The detailed process of patch merging

together. As a result, Patch Merging is more complex than pooling operations. Furthermore, pooling can result in information loss, whereas Patch Merging does not. The purpose of this module is downsampling, reducing resolution, adjusting channel numbers, and forming a hierarchical design.

Furthermore, in the Swin Transformer, self-attention is performed within small windows. These windows, composed of patches, differ from those used in Vision Transformers (ViT) and are relatively independent. For example, in a 4x downsampling scenario, the feature map is divided into multiple disjointed window regions, and Multi-Head Self-Attention is only applied within each window patch. Since the windows are disjointed, there arises the challenge of how to transmit information and learn multi-scale information, which necessitates the use of patch merging. In simple terms, patch merging combines small window patches into larger ones to increase receptive field size, as illustrated in Fig. 7.

Assuming that the input Patch Merging is a single channel feature map of 4×4 size, Patch Merging will divide each 2×2 nearby pixel into a patch and then combine the same position (same color) pixels in each patch to produce four feature maps of half the length and width. The four feature maps are then concatenated along the depth dimension and normalized using a LayerNorm layer. Lastly, the number of channels in the feature map is halved by making a linear change in the depth direction of the feature map via a fully connected layer. After each Patch Merging process, the feature map's height and breadth will be halved, while the number of channels will be doubled.

4 Experiments

4.1 Datasets

We test the effectiveness of Swin-Fusion using five distinct datasets containing diverse daily activities (drinking, jumping), fine-grained human behaviors (driver activity), and human-object interactions (playing a musical instrument). Suppose the official dataset does not

Table 1 Statistics for the five datasets used in our experiments

Dataset	Class	Train/test	Total
Li-6	6	240/120	360
PPMI-24	24	2400/2400	4800
Stanford-40	40	7640/1892	9532
AUC-V1	10	12,977/4331	17308
AUC-V2	10	12,555/1923	14478

contain a training set and test set. In that case, we randomly partition the dataset in a ratio of 8:2. these datasets vary in size from small 360, medium 9k, to large 17k, as shown in Table 1.

4.1.1 Li's Action(Li-6)

This small still image dataset has six action categories: playing guitar, riding a horse, shooting, phoning, riding a bicycle, and jogging. There are 60 images for each action category, for a total of 360 images. For each action category, 40 images are randomly picked for training and 20 for testing. Each category's images are meticulously cropped and the same size.

4.1.2 People Playing Musical Instruments (PPMI-24)

PPMI-24 is a challenging dataset with 24 distinct behavior classes and 4800 images. We follow the official divisions for training and testing, with 2400 images for training and 2400 images for testing. These 24 distinct behavior classes correlate to 12 musical instruments, each of which includes holding and playing the instrument. Backgrounds and occlusions in the PPMI-24 dataset are more intricate than those in the Stanford-40 dataset.

4.1.3 Stanford-40 Action(Stanford-40)

The dataset comprises 9532 still images depicting 40 distinct human actions, including jumping, drinking, and fishing. Since there is no defined training and testing set for this dataset, we randomly selected 80% of the photos as the training set, totaling 7640, and 20% as the testing set, totaling 1892. The action images in the Stanford-40 dataset were collected from various aspects of life, with the behavioral individuals located predominantly in the center of the images and with good image quality.

4.1.4 Distracted Driver V1 (AUC-V1)

AUC-V1 is a fine-grained behavior recognition dataset with a total of 17308 pictures, of which 12977 were used for training and 4331 were used for testing. The AUC-V1 dataset consists of 31 individuals from seven nations, including the United States, Palestine, and Egypt. Ten driving activities included: conversing with passengers, driving safely, answering the phone on the left side, and drinking.

4.1.5 Distracted Driver V2 (AUC-V2)

AUC-V2 is an upgraded version of the preceding dataset, containing 14,478 photos and ten driving activity categories. The images also depict 44 drivers (29 men and 15 women) from seven different countries. Additionally, we followed the initial training and testing divisions, which included 12,555 training images from 38 drivers and 1923 test images from the remaining six drivers.

4.2 Experimental Settings

The Swin-Transformer comes in four sizes: Swin Tiny (Swin-T), Swin Small (Swin-S), Swin Base (Swin-B), and Swin Large (Swin-L). We select Swin-L as the model's backbone network and utilize the pre-trained weights on the ImageNet-22K dataset, which consists of 22K categories and 14.2 million images. In Swin-L, the input image size is 384×384, the window size is set to 12, the number of channels in the feature map of the first stage is 192, and the number of Swin-Transformer blocks in each stage and the number of heads in the multi-head attention module are respectively set to [2,2,18,2] and [6,12,24,48]. Four NVIDIA 1080 GPUs, each with 8GB of video memory, are utilized. We employ the AdamW optimizer with a weight decay of 5e-2, an epoch number of 40, a batch size of 16, an initial learning rate of 1e-5, and the Trivial Augment data enhancement method.

4.3 Experimental Results

To better assess the effectiveness of Swin-Fusion, we tested on a total of five still image-based HAR datasets and compared the test results with the models since 2018. Most of these methods are based on improved CNNs, and no transformer-based method exists. Table 2 displays the results.

Because the Li-6 dataset contains only 360 images, few publications have utilized this behavior recognition dataset in recent years, and only one DELVS3 with a maximum accuracy of 99.17% was discovered. The other four datasets have been utilized more frequently in recent years, and four are listed below. We also list the results of Swin-Transformer without the feature fusion module, and it should be noted that Swin-L is the backbone network of Swin-Transformer. Swin-Transformer and Swin-Fusion have achieved 100% accuracy on a tiny dataset, such as Li-6, as shown in the table. In the actual test, 100% accuracy was achieved in only three epochs, demonstrating the superiority of the Swin-Transformer network with pre-trained weights. On the AUC-V2 dataset, Swin-Fusion with the feature fusion module can improve accuracy by up to 1.33% compared to Swin-Transformer. According to the final experimental results, Swin-Fusion's transformer-based model can achieve more excellent performance in the still image-based HAR challenge.

Figure 8 depicts the heat maps generated with Grad-CAM [43] to determine if Swin-Fusion focuses on the critical feature information. We present four sorts of behavior: applauding, waving hands, riding a horse, and using a computer. The graphs demonstrate that Swin-Fusion identifies well between behavioral acts and other irrelevant content and captures crucial behavioral characteristics. As demonstrated by Grad-CAM, Swin-Fusion is successful for behavior recognition based on still images.

Table 2 Accuracy comparison of Swin-Fusion with other methods on 5 datasets

Dataset	Model / Acc(%)					
Li-6	–	–	DELWO3[15]	DELVS3[15]	Swin-transformer	Swin-fusion
	–	–	98.33	99.17	100	100
PPMI-24	Resnet34+DSF-Net[32]	NAS-Net[33]	AG-Net[34]	RAN[35]	Swin-transformer	Swin-fusion
	72.36	82.3	98.2	98.63	97.69	98.26
Stanford-40	AlexNet+α-pooling[36]	NAS-Net[8]	MCA-Net[7]	RAN[35]	Swin-transformer	Swin-fusion
	87.7	93.17	93.76	97.43	96.24	96.72
AUC-V1	CTA-Net[37]	Modelintegration[38]	AlexNet+HOG [39]	DenseNet+Latent pose[40]	Swin-transformer	Swin-Fusion
	84.09	92.36	93.19	94.2	94.41	94.83
AUC-V2	InceptionV3[36]	Inception V3+Feature fusion[41]	CTA-Net[37]	C-SLSTM[42]	Swin-transformer	Swin-fusion
	90.07	90.38	92.5	92.7	92.33	93.66

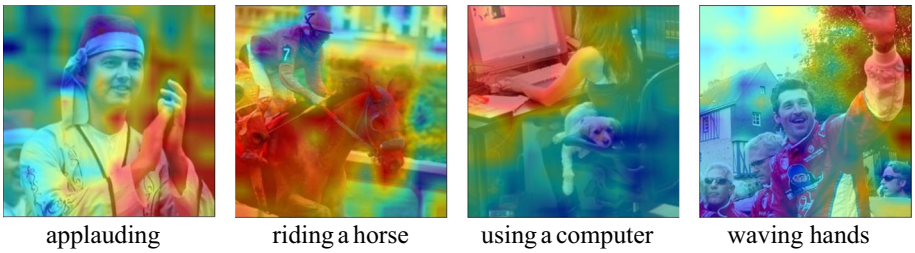

Fig. 8 Grad-CAM effect on still images

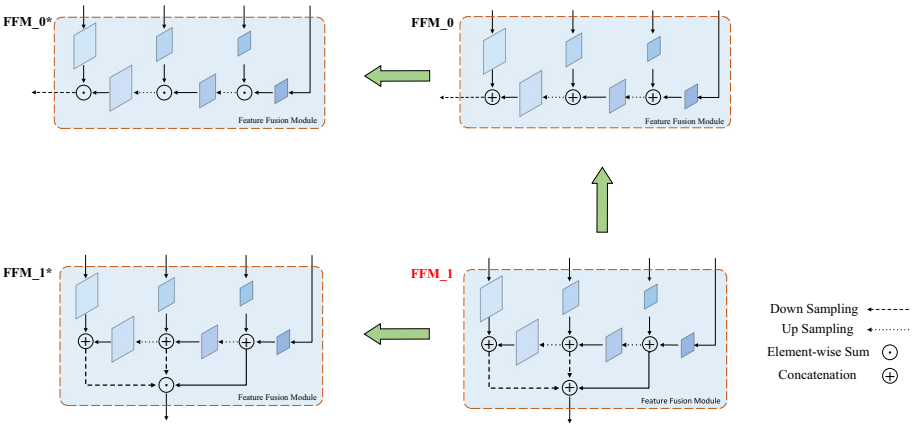

Fig. 9 Four feature fusion models using different fusion methods. (* denotes FFM using element-wise sum)

4.4 Ablation Study

4.4.1 Feature Fusion Module

Different Fusion Methods

Figure 9 shows that we have constructed four feature fusion modules, with FFM_1 having the most excellent enhancement effect. Two primary tactics were employed in the design: element-wise sum and concatenation. The element-wise sum is the superposition of pixels, in which the quantity of information in a single feature map rises. At the same time, the number of channels in the image remains constant. The FFM using element-wise sum will be marked with * at the end of the name. Concatenation increases the number of channels in an image but does not enhance the amount of information in a single feature map.

Thus, concatenation retains more feature information, but the element-wise sum is somewhat less expensive to compute. According to these two strategies, we extract and fuse the feature maps of the last three stages of the Swin-L model. By designing multiple feature fusion methods and conducting many experiments to make them applicable to multiple datasets, we apply FFM_1, the method with the best performance, to Swin-Fusion.

Table 3 illustrates the results of the four feature fusion modules on each dataset when we modify the feature fusion module without modifying the original Swin-L model's other parameter settings. It is worth noting that, for example, in FFM_0*, to ensure that the final number of channels is 1536, the number of channels will be increased through convolution

Table 3 The enhancement effect of four feature fusion modules using different fusion methods

Method	Params	GFLOPs	PPMI-24	Stanford-40	AUC-V1	AUC-V2
FFM_0	2.069M	1.452G	+ 0.20	+ 0.27	+0.25	+ 0.68
FFM_0*	8.256M	3.49G	+ 0.06	+ 0.16	+0.19	+ 0.17
FFM_1	3.25M	1.622G	**+ 0.57**	**+ 0.48**	**+ 0.42**	**+ 1.33**
FFM_1*	9.066M	3.745G	+ 0.26	+ 0.33	+ 0.31	+ 0.72

The bold data indicate the FFM with the best improvement effect * denotes FFM using element-wise sum

Table 4 The enhancement effect of four feature fusion modules

Method	Params	GFLOPs	PPMI-24	Stanford-40	AUC-V1	AUC-V2
FFM_0	2.069M	1.452G	+ 0.20	+ 0.27	+0.25	+0.68
FFM_1	3.25M	1.622G	**+ 0.57**	**+ 0.48**	**+ 0.42**	**+ 1.33**
FFM_2	8.265M	4.17G	+ 0.13	+ 0.21	+ 0.29	+ 0.62
FFM_3	7.673M	4.085G	−0.11	+ 0.09	−0.12	+ 0.21

The bold data indicate the FFM with the best improvement effect

kernel convolution, which is why the parameter amount of FFM_0* is so large. The table results show that FFM utilizing concatenation performs better than FFM using element-wise sum. Although the element-wise sum approach can enhance the amount of information in each feature map, it also introduces interference, rendering the feature fusion inefficient. In contrast, the concatenation method concats the channels to preserve all the information. In the case of FFM_0* and FFM_0, the results of FFM_0 using the concatenation approach will be superior to those of FFM_0* using the element-wise sum method, proving the preceding statement. Using the concatenation method, we fuse the features derived from FFM_0 to get FFM_1 with the best lifting effect. FFM_1 utilizes low-level features with high resolution and high-level features with high semantic information. In addition, it does not rely solely on the last layer of features for prediction by FFM_1 but combines the features at various scales of these three stages to create the optimal fusion effect.

Different Fusion Positions

As shown in Fig. 10, we tried different combinations based on FFM_1, such as FFM_2 and FFM_3. Nevertheless, the result is unsatisfactory, as shown in Table 4, indicating that only fusing the characteristics of the last three stages is the most effective.

It is beneficial to obtain more accurate features when adjacent features are fused. At the same time, too much reuse of features can lead to a decrease in accuracy on some datasets. By comparing these 6 FFMs, we use the best FFM_1 for Swin-Fusion.

Figure 11 reflects the relationship between Params, FLOPS, and increased accuracy of six feature fusion modules on four action recognition datasets through scatter plots. The figure shows that the accuracy is not directly proportional to the number of parameters or FLOPS. Therefore, how to design the feature fusion model is very important. In the design process, the number of channels should be considered. If the number of channels is forced to increase in the end, the accuracy will decrease, and the number of parameters will increase.

4.4.2 Shifted Windows Multi-head Self-attention (SW-MSA)

Table 5 shows the ablation experiments of SW-MSA, an essential module in the model. In the table, "w/o shifting" refers to the non-shifted self-attention module. Swin-Fusion with

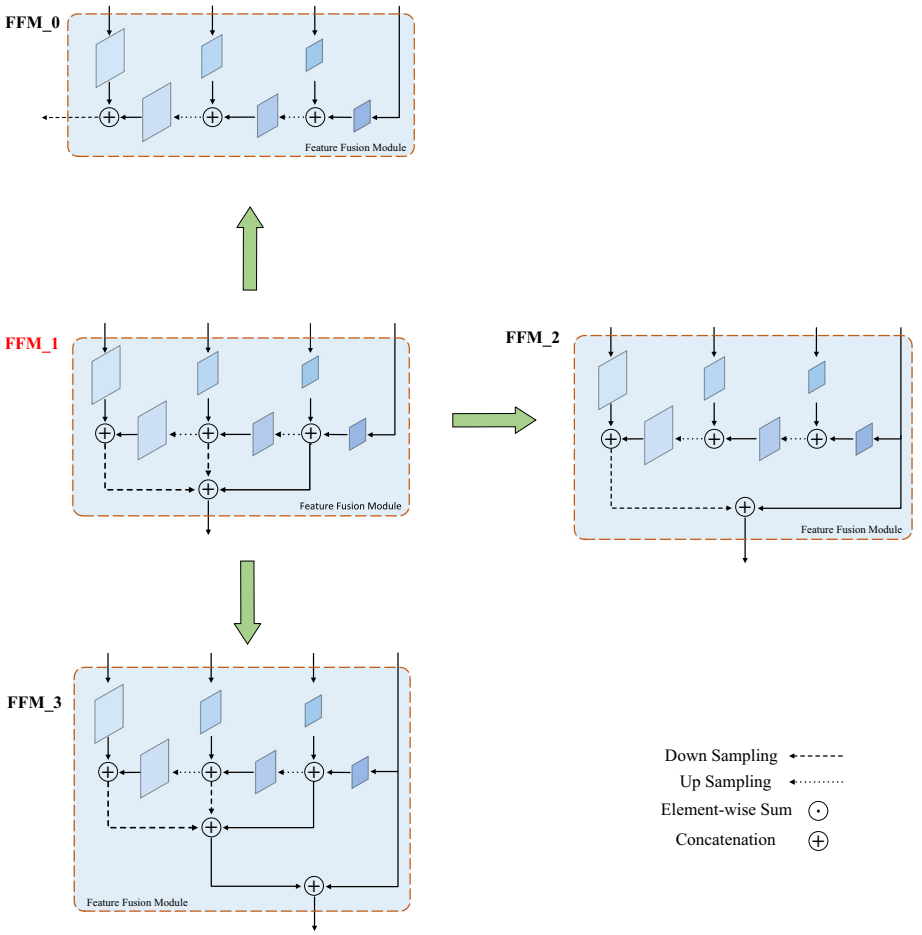

Fig. 10 Four feature fusion modules using different fusion positions

Table 5 Ablation experiments on the SW-MSA module were conducted using Swin-Fusion on five different datasets

	PPMI-24	Stanford-40	AUC-V1	AUC-V2
W/o shifting	97.33	95.85	94.01	92.63
Shifted windows	98.26	96.72	94.83	93.66

Swin-L as the base model and an input size of 384×384 was compared on four different datasets. The results show that utilizing the SW-MSA module leads to an approximate 1% improvement in accuracy, indicating the effectiveness of using shifted windows.

4.4.3 Input Resolution and Model Size

We tested the effects of various input resolutions and model sizes on the experimental outcomes of the Swin-Fusion model, with the input image resolution set to 224×224 and

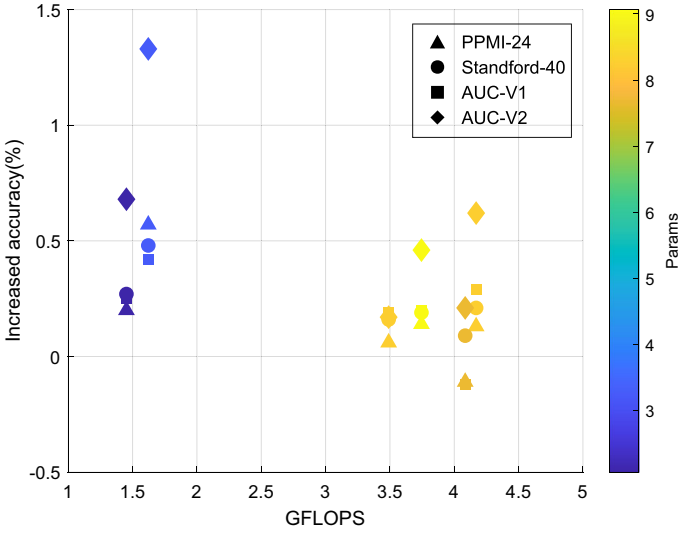

Fig. 11 The relationship between Params, FLOPS, and improved accuracy of the six feature fusion modules on four still image-based human action recognition datasets

Table 6 Results of different input resolution and model size

Backbone	Input size	Params	GFLOPs	PPMI-24	Stanford-40	AUC-V1	AUC-V2
Swin-B	224×224	90.133M	15.685G	96.75	95.45	93.77	92.15
Swin-B	384×384	90.133M	46.174G	97.54	95.91	94.32	93.04
Swin-L	224×224	202.635M	35.242G	97.31	95.88	94.09	92.92
Swin-L	384×384	202.635M	103.688G	98.26	96.72	94.83	93.66

384×384, the model size Swin-B and Swin-L, respectively. As shown in Table 6, except for the Li-6 dataset, the larger the input resolution of the image, the higher the final classification accuracy can be improved by 0.74% on average for the same model size.

Moreover, as the size of the Swin-Fusion model increases, the final output improves by an average of 0.6% with the same image input resolution. The performance improvement is insignificant, even though the number of parameters in Swin-L is more than twice that of Swin-B. While Swin-L performs the best at a resolution of 384×384, the increase in GFLOPS is very apparent. Swin-B with a higher resolution should be used if computation cost is a higher priority.

5 Conclusion

CNNs have been frequently used to recognize human action from still images. In this study, we investigate the performance of the Swin-Transformer in recognizing human actions from still images. Without modifying the original Swin-Transformer model's backbone, a simple but effective feature fusion module is introduced, enabling the model to accomplish more effective action recognization by adopting a lateral connection and top-down pathway to

merge features from various stages. Swin-Fusion was evaluated using pre-trained weights on five still image-based action recognition datasets, including the Li's action dataset, the Stanford-40 dataset, the PPMI-24 dataset, the AUC-V1 dataset, and the AUC-V2 dataset. Our model delivers competitive performance compared to improved CNNs approaches developed in the last five years. Although our model performs better than prior methods, the Swin-L version of Swin-Fusion is still enormous, and additional study is required to lower its computing cost. We plan to use other lightweight transformer models combined with the optimized feature fusion module to explore more possibilities for human action recognition based on still images.

Author Contributions Tiansheng Chen: Conceptualization, Methodology, Software, Data curation, Writing– original draft, Visualization, Investigation, Validation. Lingfei Mo: Conceptualization, Writing– review & editing, Supervision.

Funding This study was funded by the National Natural Science Foundation of China [61603091, Multi-Dimensions Based Physical Activity Assessment for the Human Daily Life] and the 2021 Blue Project for University of Jiangsu Province.

Declarations

Conflict of interest The authors declare no competing interests.

References

1. Lin J, Gan C, Han S (2019) Tsm: Temporal shift module for efficient video understanding. In: Proceedings of the IEEE/CVF international conference on computer vision, 7083–7093
2. Li K, Wang Y, Gao P, Song G, Liu Y, Li H, Qiao Y (2022) Uniformer: Unified transformer for efficient spatiotemporal representation learning. arXiv preprint arXiv:2201.04676
3. Girdhar R, Singh M, Ravi N, van der Maaten L, Joulin A, Misra I (2022) Omnivore: A single model for many visual modalities. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 16102–16112
4. Zhang J, Yang J, Yu J, Fan J (2022) Semisupervised image classification by mutual learning of multiple self-supervised models. Int J Intell Syst 37(5):3117–3141
5. Qi T, Xu Y, Quan Y, Wang Y, Ling H (2017) Image-based action recognition using hint-enhanced deep neural networks. Neurocomputing 267:475–488
6. Lavinia Y, Vo HH, Verma A (2016) Fusion based deep cnn for improved large-scale image action recognition. In: 2016 IEEE international symposium on multimedia (ISM), 609–614. IEEE
7. Hirooka K, Hasan MAM, Shin J, Srizon AY (2022) Ensembled transfer learning based multichannel attention networks for human activity recognition in still images. IEEE Access 10:47051–47062
8. Mohammadi S, Majelan SG, Shokouhi SB (2019) Ensembles of deep neural networks for action recognition in still images. In: 2019 9th international conference on computer and knowledge engineering (ICCKE), 315–318. IEEE
9. Chong Z, Mo L (2022) St-vton: self-supervised vision transformer for image-based virtual try-on. Image Vis Comput 127:104568
10. Yu J, Li J, Yu Z, Huang Q (2019) Multimodal transformer with multi-view visual representation for image captioning. IEEE Trans Circuits Syst Video Technol 30(12):4467–4480
11. Zhang J, Cao Y, Wu Q (2021) Vector of locally and adaptively aggregated descriptors for image feature representation. Pattern Recogn 116:107952
12. Csurka G, Dance C, Fan L, Willamowski J, Bray C (2004) Visual categorization with bags of keypoints. In: Workshop on statistical learning in computer vision, ECCV, vol 1, 1–2. Prague
13. Ikizler N, Cinbis RG, Pehlivan S, Duygulu P (2008) Recognizing actions from still images. In: 2008 19th international conference on pattern recognition, pp 1–4. IEEE
14. Yao B, Khosla A, Fei-Fei L (2011) Combining randomization and discrimination for fine-grained image categorization. In: CVPR 2011, pp 1577–1584. IEEE

15. Yu X, Zhang Z, Wu L, Pang W, Chen H, Yu Z, Li B (2020) Deep ensemble learning for human action recognition in still images. Complexity **2020**
16. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, 770–778
17. Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V, Rabinovich A (2015) Going deeper with convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition, 1–9
18. Sreela S, Idicula SM (2018) Action recognition in still images using residual neural network features. Procedia Comput. Sci. 143:563–569
19. Gkioxari G, Girshick R, Malik J (2015) Contextual action recognition with r* cnn. In: Proceedings of the IEEE international conference on computer vision, 1080–1088
20. Zhang Y, Cheng L, Wu J, Cai J, Do MN, Lu J (2016) Action recognition in still images with minimum annotation efforts. IEEE Trans Image Process 25(11):5479–5490
21. Dosovitskiy A, Beyer L, Kolesnikov A, Weissenborn D, Zhai X, Unterthiner T, Dehghani M, Minderer M, Heigold G, Gelly S, et al (2020) An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929
22. Touvron H, Cord M, Douze M, Massa F, Sablayrolles A, Jégou H (2021) Training data-efficient image transformers and distillation through attention. In: International conference on machine learning, 10347–10357. PMLR
23. Yu W, Luo M, Zhou P, Si C, Zhou Y, Wang X, Feng J, Yan S (2022) Metaformer is actually what you need for vision. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 10819–10829
24. Li Y, Yuan G, Wen Y, Hu E, Evangelidis G, Tulyakov S, Wang Y, Ren J (2022) Efficientformer: vision transformers at mobilenet speed. arXiv preprint arXiv:2206.01191
25. Liu Z, Lin Y, Cao Y, Hu H, Wei Y, Zhang Z, Lin S, Guo B (2021) Swin transformer: Hierarchical vision transformer using shifted windows. In: Proceedings of the IEEE/CVF international conference on computer vision, 10012–10022
26. Cruz-Mota J, Bogdanova I, Paquier B, Bierlaire M, Thiran J-P (2012) Scale invariant feature transform on the sphere: theory and applications. Int J Comput Vis. 98(2):217–241
27. Dalal N, Triggs B (2005) Histograms of oriented gradients for human detection. In: 2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05), vol 1, 886–893. IEEE
28. Long J, Shelhamer E, Darrell T (2015) Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, 3431–3440
29. Hariharan B, Arbeláez P, Girshick R, Malik J (2015) Hypercolumns for object segmentation and fine-grained localization. In: Proceedings of the IEEE conference on computer vision and pattern recognition, 447–456
30. Ghiasi G, Fowlkes CC (2016) Laplacian pyramid reconstruction and refinement for semantic segmentation. In: European conference on computer vision, 519–534. Springer
31. Lin T-Y, Dollár P, Girshick R, He K, Hariharan B, Belongie S (2017) Feature pyramid networks for object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition, 2117–2125
32. Li Z, Ge Y, Feng J, Qin X, Yu J, Yu H (2020) Deep selective feature learning for action recognition. In: 2020 IEEE international conference on multimedia and expo (ICME), 1–6. IEEE
33. Li R, Liu Z, Tan J (2018) Reassessing hierarchical representation for action recognition in still images. IEEE Access 6:61386–61400
34. Bera A, Wharton Z, Liu Y, Bessis N, Behera A (2021) Attend and guide (ag-net): a keypoints-driven attention-based deep network for image recognition. IEEE Trans Image Process 30:3691–3704
35. Behera A, Wharton Z, Liu Y, Ghahremani M, Kumar S, Bessis N (2020) Regional attention network (ran) for head pose and fine-grained gesture recognition. IEEE Trans Affect Comput. https://doi.org/10.1109/TAFFC.2020.3031841
36. Eraqi HM, Abouelnaga Y, Saad MH, Moustafa MN (2019) Driver distraction identification with an ensemble of convolutional neural networks. J Adv Transp. https://doi.org/10.1155/2019/4125865
37. Wharton Z, Behera A, Liu Y, Bessis N (2021) Coarse temporal attention network (cta-net) for driver's activity recognition. In: Proceedings of the IEEE/CVF winter conference on applications of computer vision, 1279–1289
38. Alotaibi M, Alotaibi B (2020) Distracted driver classification using deep learning. SIViP 14(3):617–624
39. Arefin MR, Makhmudkhujaev F, Chae O, Kim J (2019) Aggregating cnn and hog features for real-time distracted driver detection. In: 2019 IEEE international conference on consumer electronics (ICCE), 1–3. IEEE

40. Behera A, Keidel AH (2018) Latent body-pose guided densenet for recognizing driver's fine-grained secondary activities. In: 2018 15th IEEE international conference on advanced video and signal based surveillance (AVSS), 1–6. IEEE
41. Wu M, Zhang X, Shen L, Yu H (2021) Pose-aware multi-feature fusion network for driver distraction recognition. In: 2020 25th international conference on pattern recognition (ICPR), 1228–1235. IEEE
42. Mase JM, Chapman P, Figueredo GP, Torres MT (2020) A hybrid deep learning approach for driver distraction detection. In: 2020 international conference on information and communication technology convergence (ICTC), 1–6. IEEE
43. Selvaraju RR, Cogswell M, Das A, Vedantam R, Parikh D, Batra D (2017) Grad-cam: Visual explanations from deep networks via gradient-based localization. In: Proceedings of the IEEE international conference on computer vision, 618–626