electronics

MDPI

Article

Optimization of Action Recognition Model Based on Multi-Task Learning and Boundary Gradient

Yiming Xu, Fangjie Zhou, Li Wang *, Wei Peng and Kai Zhang

College of Electrical Engineering, Nantong University, Nantong 226019, China; yimingx@ntu.edu.cn (Y.X.); 1912310014@stmail.ntu.edu.cn (F.Z.); 2011310022@stmail.ntu.edu.cn (W.P.); 2011310011@stmail.ntu.edu.cn (K.Z.)
* Correspondence: lwee@ntu.edu.cn; Tel.: +86-0513-85012601

Abstract: Recently, people's demand for action recognition has extended from the initial high classification accuracy to the high accuracy of the temporal action detection. It is challenging to meet the two requirements simultaneously. The key to behavior recognition lies in the quantity and quality of the extracted features. In this paper, a two-stream convolutional network is used. A three-dimensional convolutional neural network (3D-CNN) is used to extract spatiotemporal features from the consecutive frames. A two-dimensional convolutional neural network (2D-CNN) is used to extract spatial features from the key-frames. The integration of the two networks is excellent for improving the model's accuracy and can complete the task of distinguishing the start–stop frame. In this paper, a multi-scale feature extraction method is presented to extract more abundant feature information. At the same time, a multi-task learning model is introduced. It can further improve the accuracy of classification via sharing the data between multiple tasks. The experimental result shows that the accuracy of the modified model is improved by 10%. Meanwhile, we propose the confidence gradient, which can optimize the distinguishing method of the start–stop frame to improve the temporal action detection accuracy. The experimental result shows that the accuracy has been enhanced by 11%.

Keywords: multi-scale; multi-task learning; temporal action detection; confidence gradient

Citation: Xu, Y.; Zhou, F.; Wang, L.; Peng,W.; Zhang, K. Optimization of Action Recognition Model Based on Multi-Task Learning and Boundary Gradient. *Electronics* **2021**, *10*, 2380. https://doi.org/10.3390/electronics10192380

Academic Editor: Manohar Das

Received: 2 September 2021
Accepted: 25 September 2021
Published: 29 September 2021

1. Introduction

The rapid development of artificial intelligence has greatly promoted the quality of people's life. Action-recognition-based vision is an essential branch in the field of artificial intelligence. It plays a vital role in human–computer interaction, security and the smart home [1,2]. At present, action recognition is mainly based on two types of network, Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN). As a special RNN, Long Short-Term Memory (LSTM) has great advantages in long video action detection. Ref. [3] was based on the LSTM network. It realized that the information of joints in the action sequence is selectively realised with the help of global context information. Ref. [4] introduced a new gating mechanism based on Spatio–Temporal LSTM. It can stably learn the sequential of input data, which is beneficial for improving the accuracy. Many other LSTM-based methods have achieved high accuracy [5,6]. CNN has advantages in terms of computing speed and can form a good balance between accuracy and effectiveness in short videos. In order to take advantage of CNN's strengths and make up for its shortcomings, CNN has progressed from the previous 2D-CNN and 3D-CNN to the current two-stream CNN [7–10]. This has achieved good results. However, two-stream CNN still has space for growth in terms of its recognition accuracy.

While obtaining high behavior-recognition accuracy, there is also a high accuracy requirement for the start–end time of the action in some scenarios. In order to realize the temporal detection of the action, it is necessary to first find the proposal, then classify the proposal, and finally obtain the regression boundary. The methods used to search for the proposals are divided into three main types. The first is based on a simple sliding window [11]. It can find the most complete proposals, but there is a large amount of redundancy. The second is based on individual classification to the video frames. It divides the adjacent ones within the same category into a group [12]. It prevents noise interference by setting some thresholds. This method is flexible for the boundary, but the proposals may be missed due to incorrect classification. While the calculation efficiency is improved, the accuracy is reduced. The third is based on dividing the video into fixed-size units [13]. It takes N frames as a group. It learns a feature in C3D, and then takes one or more groups as the central anchor unit. It performs unit-level regression on proposals that extend different lengths to the same length. This method achieves a good balance between accuracy and efficiency. However, there is still room to improve the accuracy of temporal detection. Inspired by the above research, this paper designs a model based on a two-stream CNN, which improves the accuracy of action recognition and optimizes temporal detection. The main contributions of this paper are as follows:

1. The accuracy of action classification is improved through the fusion of spatiotemporal multi-scale features and the sharing of information between multiple tasks;
2. The confidence gradient is proposed. We improve the accuracy of temporal action detection through the sensitive boundary confidence gradient;
3. A high-resolution dataset of multi-person action recognition under the light interference and foreign object interference is made.

2. Related Work

Researchers have spent much effort to improve the accuracy of action recognition. Ref. [14] expanded 2D-CNN from the spatial dimension to the time dimension and designed a 3D-CNN called C3D. Ref. [15] was the first two-stream CNN structure. The researcher used single frame and multi-frame optical flow as input to realize action recognition at spatial and temporal dimensions. He achieved good results with limited training data, and made significant progress in the implementation of deep-learning-based action classification. Ref. [16] explored the fusion method of the output characteristics of two-stream networks. They found that fusing spatio–temporal networks in the convolution layer can greatly reduce the number of parameters without causing performance degradation. Ref. [17] further discussed the fusion of spatio–temporal features and put forward more improved structures. The use of a two-stream network for action recognition has advantages in terms of efficiency. However, the two-stream architecture handles action and appearance separately in each branch, which may lead to overfitting on appearance information. Ref. [18] designed an asymmetric attention module called yhr Motion-Attentive Transition (MAT) asymmetric attention module. This realized the close hierarchical interaction between object action and appearance and reduced overfitting.

In order to better extract the temporal and spatial features of images and videos, Refs. [19–22] used a Feature Pyramid Network (FPN) to extend feature extraction from a single layer to multiple layers. When FPN is added to the model, it can extract not only the top-level information, but also the semantic information of the bottom-level features. Refs. [23,24] extended action recognition from single-task learning to multi-task learning. The performance of the network and the accuracy of action recognition are improved by sharing the information among multiple tasks. The addition of multi-tasks increases the complexity of the model, Ref. [25] proposed a cascaded multi-branch structure. This processes the video from rough to detailed, which reduces the complexity of the task.

Researchers have struggled to obtain a more precise start–end time. SCNN was used for temporal detection for the first time. The SCNN used a sliding window of different scales to find some proposals. It judges whether these proposals are actions, and then uses these proposals to determine the boundaries of the action. Finally, it obtains the start–end time of the action. TURN reduces the calculation amount of the sliding window and improves the accuracy of temporal detection through dividing the video into equal-length units and performing the unit regression. TAG pays more attention to the action content. It reduces the number of proposals and the number of calculations. TAG merges fragments to enhance the precise of temporal detection. CDC [26] samples in space and time to make up for the fine-grained loss of C3D in time. In order to distinguish between complete proposals and incomplete proposals, Ref. [27] constructed a complete structure for the time pipeline link, which can capture more fine-grained, target-specific information on the segmented objects. It improved the accuracy of temporal detection.

3. Methodology

In this paper, the action recognition task is divided into two sub-tasks: action classification and temporal action detection. Firstly, we completed the preliminary positioning of action space and time based on You Only Watch Once (YOWO) [28]. Then, we introduced a multi-scale feature extraction module to extract features of different depth layers and extract different sizes' features on the same layer, improving the accuracy of classification. The multitask learning module is also adopted; more feature information is sparsely shared by digging the relationship between multiple tasks. The method of confidence gradient is introduced into the judgment of time-sensitive boundaries [29]. Under the condition of the start–stop frame being initially determined, the start–stop frame is accurately located, which leads to higher accuracy for the temporal action detection.

3.1. *Action Classification*

A 3D convolutional network is used to extract spatiotemporal features and a 2D convolutional network is used to extract spatial features. The 3D feature data are integrated with the 2D feature data after dimension transformation. Then, the features are fused to complete the task of action classification. The multi-scale feature extraction is used to increase the receptive field at the time of feature extraction. The method of multitask learning is also adopted in the training process. All the detected objects are sent into the multitask classifier to obtain the action-classification model under multitask learning. The recognition accuracy of the model is improved by sharing the information between multiple tasks. The separable depth convolution is used to reduce the parameters of the model, which can enhance the model's operation efficiency. The overall framework of action recognition is shown in Figure 1.

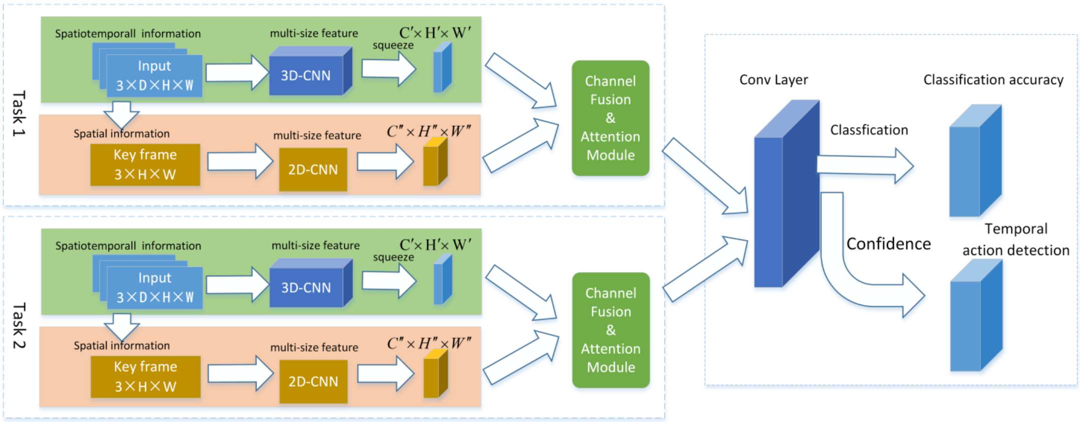


Figure 1. Overall framework of action recognition.

3.1.1. Feature Extraction and Fusion

3D-CNN is good at capturing dynamic information, and 2D-CNN can capture static position information with a small amount of calculation, so the paper adopts the method of merging 3D-CNN and 2D-CNN. The 3D Resnext-101 network is used as the fundamental backbone of the 3D-CNN. The input of the 3D-CNN is a series of consecutive frames. The shape of this is $[C \times D \times H \times W]$. $C = 3$, D represents the number of input frames, H represents the height of the input frame, and W denotes the width of the input frame. The output shape of the 3D-CNN's final conv layer is $[C' \times D' \times H' \times W']$. C' denotes the number of output channels, $D' = 1$, $H' = H/32$, $W' = W/32$. The depth dimension of the output is reduced to 1 for subsequent integration with 2D-CNN. The output is reshaped to $[C' \times H' \times W']$ after dimensional transformation. 2D-CNN is used to simultaneously extract the features of the key frame. The key frame is the last frame of each group. The shape of the input is $[C \times H \times W]$. 2D-CNN's shape of the output is $[C'' \times H' \times W']$. C'' represents the number of out channels. H' and W' have the same meaning as the 3D-CNN. Darknet-19 is applied as the fundamental backbone of the 2D-CNN in the paper.

The output of the 3D network and the 2D network is consistent in dimension through dimension transformation, so that the two feature maps can easily be integrated. Stacking along the channel is used to concatenate these features. The graph of the data processing, schematic diagram of channel fusion, and attention mechanism are shown in Figure 2.

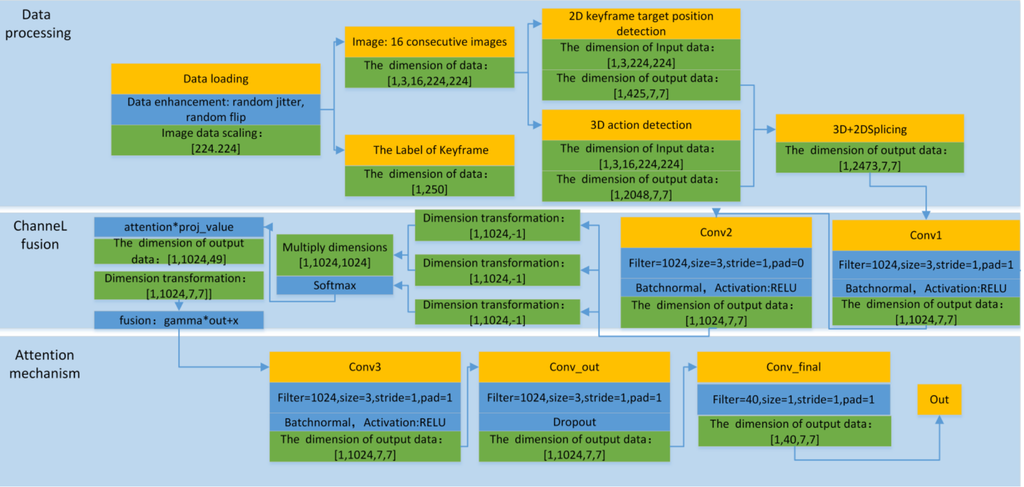

Data processing
Data loading
Data enhancement: random jitter, random flip
Image data scaling：[224.224]
Image: 16 consecutive images
The dimension of data：[1,3,16,224,224]
The Label of Keyframe
The dimension of data：[1,250]
2D keyframe target position detection
The dimension of Input data：[1,3,224,224]
The dimension of output data：[1,425,7,7]
3D action detection
The dimension of Input data：[1,3,16,224,224]
The dimension of output data：[1,2048,7,7]
3D+2DSplicing
The dimension of output data：[1,2473,7,7]
ChanneL fusion
attention*proj_value
The dimension of output data：[1,1024,49]
Dimension transformation：[1,1024,7,7]]
fusion：gamma*out+x
Multiply dimensions [1,1024,1024]
Softmax
Dimension transformation：[1,1024,-1]
Dimension transformation：[1,1024,-1]
Dimension transformation：[1,1024,-1]
Conv2
Filter=1024,size=3,stride=1,pad=0
Batchnormal，Activation:RELU
The dimension of output data：[1,1024,7,7]
Conv1
Filter=1024,size=3,stride=1,pad=1
Batchnormal，Activation:RELU
The dimension of output data：[1,1024,7,7]
Attention mechanism
Conv3
Filter=1024,size=3,stride=1,pad=1
Batchnormal，Activation:RELU
The dimension of output data：[1,1024,7,7]
Conv_out
Filter=1024,size=3,stride=1,pad=1
Dropout
The dimension of output data：[1,1024,7,7]
Conv_final
Filter=40,size=1,stride=1,pad=1
The dimension of output data：[1,40,7,7]
Out

Figure 2. The graph of the data processing, schematic diagram of channel fusion, and attention mechanism.

The information $A \in R^{(C'+C'') \times H \times W}$ cmprised of the 2D network and the 3D network is put into two convolutional layers to generate a feature map $B \in R^{C \times H \times W}$. Then, the tensor of feature map B is reshaped, and the feature vector of each channel is transformed into one dimension.

$$B \in R^{C \times H \times W} \xrightarrow{\text{vectorization}} F \in R^{C \times N} \tag{1}$$

We associate features across channels. We obtain the Gram matrix $G \in R^{C \times C}$ through $F \in R^{C \times N}$ and $F^T \in R^{N \times C}$

$$G = F \times F^T \tag{2}$$

It can be expressed as

$$G_{ij} = \sum_{k=1}^{N} F_{ik} \cdot F_{jk} \tag{3}$$

The softmax layer is used to generate a channel attention graph $M \in R^{C \times C}$ based on the Gram matrix.

$$M_{ij} = \frac{\exp(G_{ij})}{\sum_{j=1}^{C} \exp(G_{ij})} \tag{4}$$

Matrix multiplication is performed on M and F to realize the attention mapping on the original features. Then, the result is transformed into three-dimensional space with the same shape as the input tensor $R^{C \times H \times W}$.

$$F' = M \cdot F \tag{5}$$

$$F' \in R^{C\times C} \overset{\text{reshape}}{\longrightarrow} F'' \in R^{C\times H\times W} \tag{6}$$

3.1.2. Feature Pyramid Network

The method of multi-scale feature extraction is used in feature extraction. It is conducive to obtaining characteristics of different sizes and the recognition of objects with different sizes, to improve the recognition accuracy. In this paper, we use the pyramid model for both spatial and temporal feature extraction. Spatial multi-scale feature extraction consists of two parts:

1. Bottom-up process: sampling from each level up to the next-to-last level;
2. Top-down and side connection fusion process: the top-down process enlarges the top-level small feature frame by up-sampling, then, we fuse it in the upper layer.

This method not only uses the top semantic features but also the location information of the shallow features, whichis conducive to the detection of small targets. The schematic diagram is shown in Figure 3.

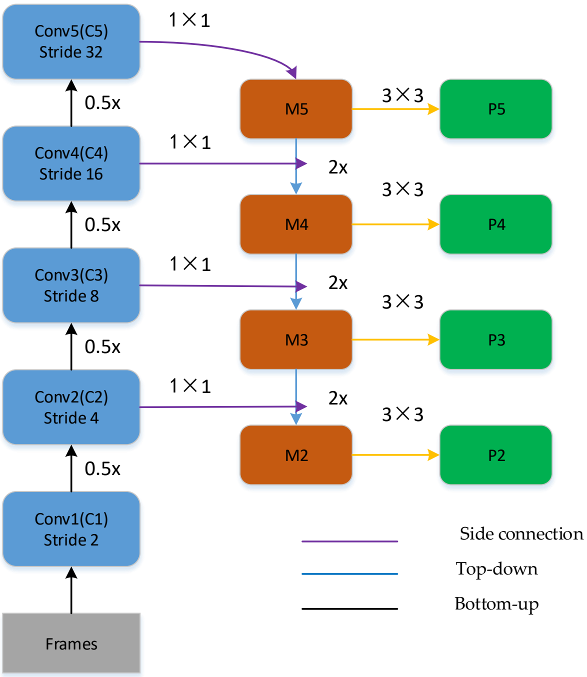


Figure 3. Spatial pyramid model.

Temporal multi-scale feature extraction is similar to spatial multi-scale feature extraction. Different scales are used for down-sampling on the time axis to obtain various features, and then fuse the various features. The schematic diagram is shown in Figure 4. The spatial modulation module aligns the features of different depths, the temporal modulation module introduces a series of sequential down-sampling factors to control different time scales, and the information flow module is adopted for the feature fusion.

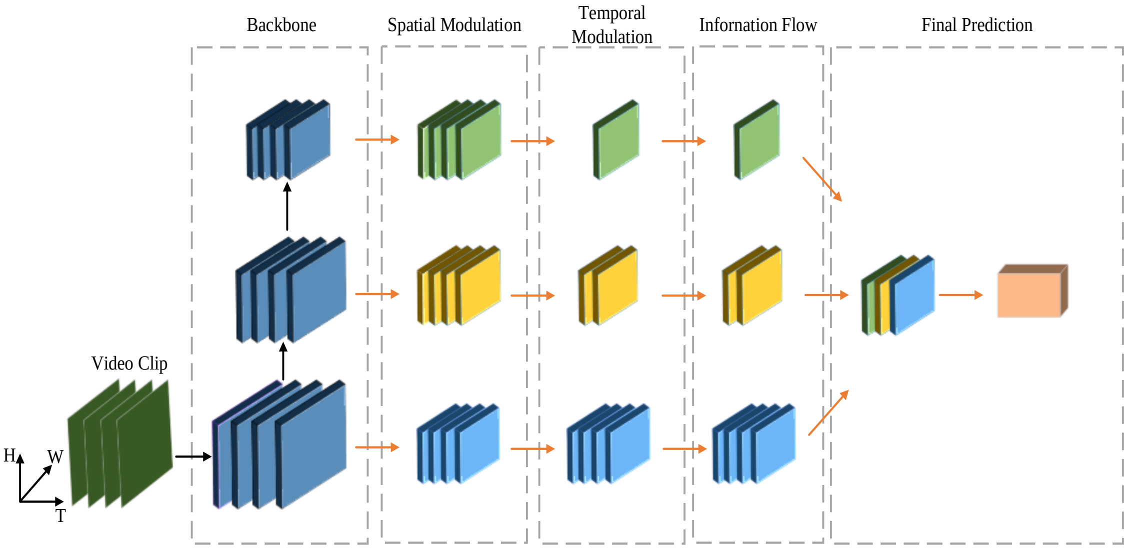


Figure 4. Temporal pyramid model.

This paper adopts multi-depth pyramid to collect features in different layers, which is conducive to obtaining richer spatial–temporal semantic information. Spatial semantic modulation is introduced to align the spatial semantics of features in the multi-depth pyramid. Spatial semantic modulation refers to performing a set of convolutions on each feature to match the spatial shape and receptive field with the top-level features. Then, the shape of the feature is aligned. At the same time, this paper introduces time rate modulation technology to control the relative differences in features in the time dimension. After that, feature aggregation can be more effectively carried out.

3.1.3. Multi-Task Learning

Multitask learning is a method that combines multiple tasks to enhance the performance of the model. The data are embedded into the same space by sharing the sparse information, and then the multi-task is jointly trained. This is a effective way of improving the accuracy of object recognition. The process is shown in Figure 5.

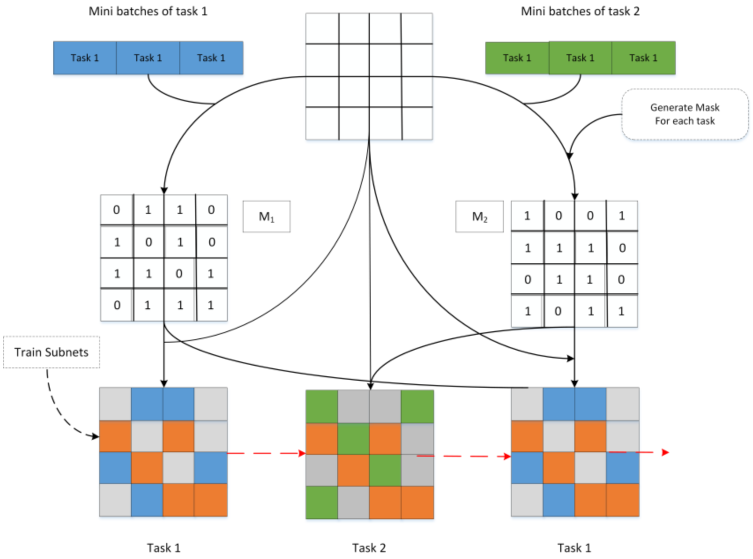


Figure 5. The multi-task model with the sparse sharing of the features.

Firstly, the data are input to the sub-networks corresponding to different tasks. Then, the parameters of the sub-networks are updated following the obtained gradient. Although only the corresponding sub-network is used to trian each task, most of the parameters of the sub-network will be shared with another task because of the correlation of tasks, so these parameters will be updated by the training data of multiple tasks to maximize the effect of multi-task learning.

In order to maximize the optimization effect of multi-task learning on the model, two multi-task regularization operations are carried out in this paper. The regularization between multiple tasks can constrain the manifold similarity of the features between tasks. The linear regression weight is used to map the features of the current task to minimize the difference in features between different tasks. Regularization is also applied to the dataset of each single task. The linear regression weight is used to map the characteristics of these datasets under a single task. It decreases the differences in the same mapped category as much as possible within a single task.

The multi-task of the paper is to extract features from the same backbone network, and then input the features to the two different branches for training. The number of layers of the experimental shared layer is equal to the number of layers of the backbone network.

3.2. Temporal Action Detection

3.2.1. The Link of Action Pipeline

Since we have the results of action classification at the frame level, the next step is to connect these detected bounding boxes and build an action pipeline in the whole video.

Assuming R_t and R_{t+1} is from the continuous frames T and T + 1, the connection score of the action category is defined as:

$$s_c(R_t, R_{t+1}) = \psi(ov) \cdot [s_c(R_t) + s_c(R_{t+1}) + \alpha \cdot s_c(R_t) \cdot s_c(R_{t+1}) + \beta \cdot ov(R_t, R_{t+1})] \tag{7}$$

$s_c(R_t), s_c(R_{t+1})$ is the score for a specific category in the region R_t and R_{t+1}. OV represents the IoU of the regions, $\psi(ov)$ =0. $\alpha \cdot s_c(R_t) \cdot s_c(R_{t+1})$ represents the significant change in the score between two consecutive frames, which can improve the performance of visual detection in the experiment.

For each action in the video, we will find the best path:

$$\bar{R}_\alpha^* = \arg\max_{\bar{R}} \frac{1}{T} \sum_{t=1}^{T-1} s_\alpha(R_t, R_{t+1}) \tag{8}$$

Among them, $\overline{R}_\alpha = [R_1, R_2, \ldots, R_T]$ represents a sequence of connected regions of the category α, Viterbi algorithm is used to optimize the result. After finding the optimal path, $\overline{R}_\alpha^*$ is removed from the region set, and Equation (7) is repeated until the region set is empty. Each path is called an action pipeline, and the score of action pipeline $\overline{R}_\alpha$ is defined as $S_\alpha\left(\overline{R}_\alpha\right) = \frac{1}{T} \sum_{t=1}^{T-1} s_\alpha(R_t, R_{t+1})$. Through the above steps, we obtain a rough action pipeline, as shown in Figure 6.

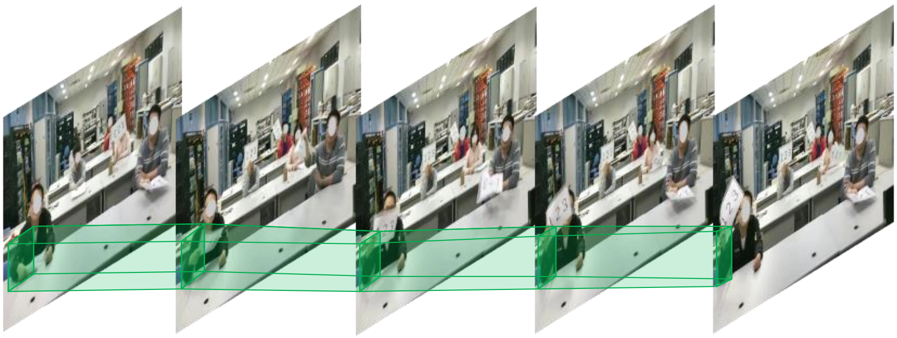


Figure 6. Pipeline linking of action.

3.2.2. Time Boundary of Action

According to the action's confidence score, generated by the model, the start action, the process action, and the end action of each position are modelled and evaluated. Thesliding window is used to slide on the sequential frames without overlapping. The confidence score of each window is used as the input to generate the successive confidence score about the start_frame, the stop_frame, and the action_frame. The corresponding confidence score of this position is obtained from the three confidence scores. At this point, we can calculate the overlap rate of the generated value with the truth value as an evaluation standard. The loss of the time boundary of action consists of three parts: action_loss, start_loss, and end_loss.

$$L_{TEM} = \lambda \cdot L_{bl}^{\text{action}} + L_{bl}^{\text{start}} + L_{bl}^{\text{end}} \tag{9}$$

The calculation formula of action-loss is as follows:

$$L = \frac{1}{l_w} \sum_{i=1}^{lw} (\alpha^{+} \cdot b_i \cdot \log(p_i) + \alpha^{-} \cdot (1 - b_i) \cdot \log(1 - p_i)) \tag{10}$$

The calculation formula of the start_loss and the end_loss is similar to the above formula. The process is shown in Figure 7.

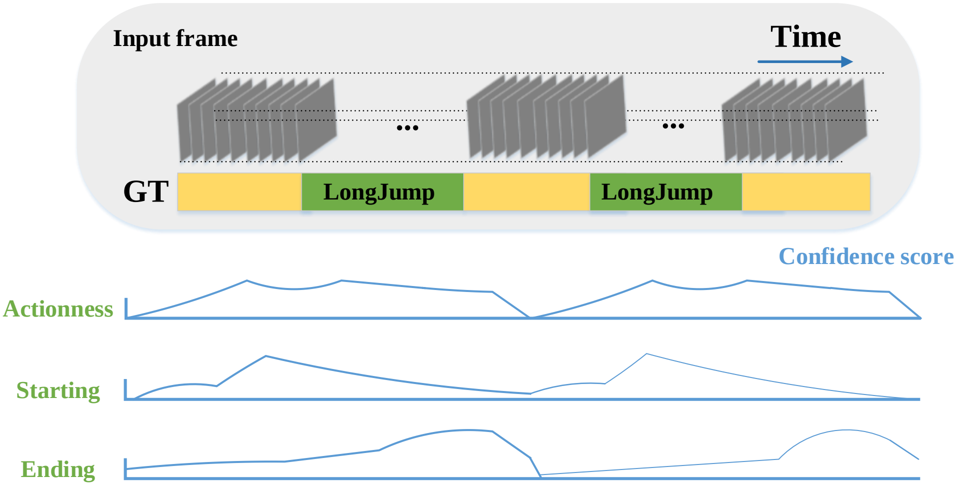


Figure 7. The judgment of time boundary.

'Actions' in the figure above refers to the confidence score of action classification, which is generated by the first part of the network. 'Starting' refers to the confidence score of the starting frame and 'ending' refers to the confidence score of the ending frame.

Aiming at the forward shift in the temporal detection caused by the start-frame's preposition in the model, we introduce the mean value of the confidence, which alleviated the problem by strengthening the accuracy of searches for the key frame. Aiming at the backward shift in temporal detection caused by the end-frame's post-position in the model, we introduce a confidence gradient to constrain the end-frame, which can alleviate the problem of the temporal detection being post-positioned.

4. Experimental Results and Analysis

4.1. Action Recognition Dataset

In this paper, two kinds of datasets are used. One of them is the public dataset UCF-24, another is the self-made dataset. Twelve action classifications are chosen in UCF-24 because multitask learning needs frames to be marked multiple times, which means that multiple task tags need to be put on the same frame; the 12 classifications are more suitable for multitask learning. The reason for building the self-made dataset is that the UCF-24 dataset is mostly a low-resolution single action in the simple background. A corresponding high-resolution dataset of multi-human action with the light interference and foreign object interference background is designed, which can expand the application scope of the model and cross-verify the effect of the model with the UCF-24 dataset.

The self-made dataset is to simulate the scene of the auction by collecting the actions of five students. The dataset is divided into three categories after being sorted out, namely, lifting a hand, touching the head, and supporting glasses. The reason for choosing the three types of actions is that they have high similarity, which can better train the model and improve the judgment ability of the model for classification. There are 30,000 collected pictures: 20,000 pictures are about lifting hands, 5000 pictures are about lifting hands, and the rest are about supporting glasses. The proportion of training set and testing set is 8:2. The self-made dataset is illustrated in Figure 8.

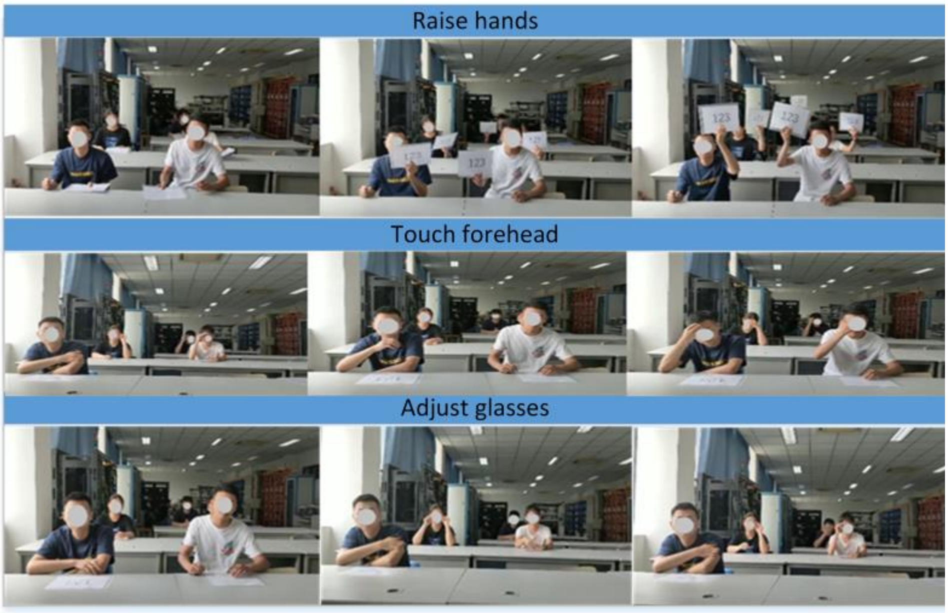


Figure 8. The self-made dataset.

4.2. Experimental Settings

In this experiment, NVIDIA GeForce RTX 3090 was used for training. During training, we use the 3D Resnext-101 network as the fundamental backbone of the 3D-CNN and Darknet-19 as the fundamental backbone of the 2D-CNN. We did not use the pre-training models of the 3D Resnext-101 and Darknet-19; we froze all parameters of the convolutional network for brand new training. The algorithm of stochastic gradient descent (SGD) is used. The momentum is set to 0.9, and the starting-learning rate is set to 0.001. A total of 16 frames are taken as a batch. The frames are normalized; the size of the input is 224 × 224. The data are strengthened by the random cutting, random left, and right flip. The learning rate decreases to 50% of the initial learning rate every 10,000 steps to alleviate the additional shock phenomenon of the model at the convergence point. We set the value of epoch as 10. When testing, we set the threshold of the bounding boxes as 0.25 and set the non-maximum threshold as 0.5 to delete and select suitable bounding boxes.

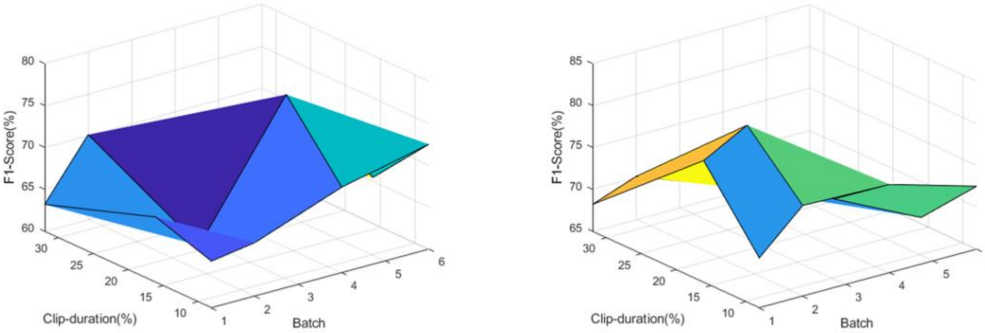
The relationship between the setting of Batch value, clip_duration value and F1 score are shown in Figure 9. The original model is YOWO, and the improved model is the model, which introduces FPN and multitask on the basis of YOWO. From the figure of the relationship between the Batch value and Clip_duration value with F1-score, we can obtain the following conclusion.

1. The longer length of the continuous frame does not lead to a better performance. Although the more extended continuous frames contain more valuable information, there will be more interference information, and the possibility that the temporal

content information is gradually broken is increased. Therefore, different types of actions are suitable for different lengths. Short and fast actions are ideal for a shorter length, such as auction; slow-motion is ideal for a longer length, such as Taiji. Therefore, we set the value of clip duration as 16 in this experiment;

2. The setting of batch value not only affects the efficiency of operation but also the accuracy of the operation. The calculation and parameter of the improved model are about twice as much as the original model. Under a certain computing capacity, the best batch value and calculation amount are generally in an inverse ratio.

Therefore, we set the batch value as 2 when we use the improved model, and set a batch value of 4 when we use the original model.



(a) Original model (b) Our model

Figure 9. The relationship between the setting of Batch value and clip_duration value with F1_score.

4.3. Analysis of Experimental Results

In the process of training, the loss value of the multi-task adopts the method of different tasks corresponding to different weights. The loss value of multi-task in this paper is based on the maximized homoscedastic Gaussian likelihood function. The classification task conforms to the probability model; therefore, the probability function is expressed as follows:

$$p\left(y \mid f^{W}(x)\right) = \text{Softmax}\left(\frac{1}{\sigma^{2}} f^{W}(x)\right) \tag{11}$$

Its logarithmic function is expressed as: $\log\left(p\left(y \mid f^{W}(x)\right)\right)$

Since there are two outputs in the model, the total logarithmic function is expressed as:

$$p\left(y_1, y_2 \mid f^{W}(x)\right) = p\left(y_1 \mid f^{W}(x)\right) p\left(y_2 \mid f^{W}(x)\right) \tag{12}$$

Maximizing the logarithmic likelihood is equivalent to minimizing the negative logarithmic likelihood.

$$\begin{aligned} &-\log\left(p\left(y_1, y_2 \mid f^{W}(x)\right)\right) = \tfrac{1}{\sigma_1^2} - \log\left(\text{Softmax}\left(f_{y_1}^{W}(x)\right)\right) + \log(\sigma_1) \\ &+ \tfrac{1}{\sigma_2^2} - \log\left(\text{Softmax}\left(f_{y_2}^{W}(x)\right)\right) + \log(\sigma_2) \end{aligned} \tag{13}$$

If the single task damage function is expressed as $\mathcal{L}_1'(W) = -\log\left(\text{Softmax}\left(f_{y_1}^{W}(x)\right)\right)$. Then, the above formula can be expressed as:

$$\frac{1}{\sigma_1^2}\mathcal{L}_1'(W) + \frac{1}{\sigma_2^2}\mathcal{L}_2'(W) + \log(\sigma_1 \sigma_2) \tag{14}$$

In the multiple tasks, smaller variance means greater weight. This ensures that the loss value of the whole is as small as possible.

True positions (TP): the number of cases that are positive and classified as positive by the classifier. False positions (FP): the number of cases that are negative but classified as positive by the classifier; False negatives (FN): the number of positive cases that are wrongly divided into negative cases by the classifier; 'precision' is a measure of accuracy. 'Recall' is coverage surface measurement. 'F1-score' is known as comprehensive classification index, comprehensively considers the precision and recall rate.

$$precision = 1.0 \times TP/(TP + FP + eps) \tag{15}$$

$$recall = 1.0 \times TP/(TP + FN{+}eps) \tag{16}$$

$$F1\_socre = 2.0 \times precision \times recall/(precision + recall + eps) \tag{17}$$

There are many 3D backbones which can combined with darknet-19 to form a two-stream convolutional network. Which is the best? The results obtained using different 3D convolution frameworks are shown in Table 1.

Table 1. Comparison of results under the different 3D convolution frameworks.

Convolution Framework	YOWO (F1-Score)	YOWO + FPN + Multitask (F1-Score)
3D-ResNext-101	80.2	82.5
3D-ResNet-101	79.6	80.6
3D-ResNext-50	77.6	78.3
3D-ResNext-18	74.3	75.2
3D-ShuffleNetV1_2.0x	70.6	71.4
3D-ShuffleNetV2_2.0x	70.5	71.2
3D-MoblieNetV1_2.0x	66.8	67.1
3D-MoblieNetV2_1.0x	66.5	66.9

As we can see in Table 1, when 3D-ResNext-101 is selected as the fundamental backbone of the 3D-CNN, the recognition effect is the best. The result is closely related to its multiple cardinality and the ability to learn more complex features. Therefore, we chose the 3D Resnext-101 network as the fundamental backbone of the 3D-CNN and Darknet-19 as the fundamental backbone of the 2D-CNN in subsequent experiments.

When we selected the best branch of the network, we use different models to compare the performance in the UCF24 dataset when the IoU is set as 0.5. The result is shown in Table 2.

Table 2. Comparison of results on UCF24.

Method	Frame-mAP	Method	Frame-mAP
Peng w/MR	58.5	ACT	65.7
T-CNN	61.3	P3D-CTN	71.1
YOWO	74.4	YOWO + FPN	75.2
YOWO + Multi-task	75.3	YOWO + FPN + Multi-task	75.9

It can be seen from Table 2, that our method has a greater advantage in terms of accuracy than other methods. The accuracy was also improved compared with pure YOWO, which shows the effectiveness of FPN and multi-task. However, the improvement effect is not particularly obvious. The main reason is that only a part of the actions on the UCF24 dataset are suitable for multi-task learning, and the dataset which is suitable for multi-task learning also has the problem of being mislabelled due to low pixels. Therefore, corresponding to the UCF24 dataset with single-person movement in the low-resolution simple background, a high-resolution action dataset with multi-person movement in its light interference and foreign object interference is made to cross-verify the model. This dataset contains three actions, namely Raise hands, Adjust glasses, and Touch forehead. The result of the comparison under the different models is shown in Table 3.

Table 3. Comparison of results on self-made dataset.

Actions	YOWO Precision	Recall	F1	YOWO + FPN Precision	Recall	F1
Raise hands	68.03	72.50	70.26	71.56	78.53	75.05
Adjust glasses	65.19	69.32	67.25	67.13	76.56	71.85
Touch forehead	66.08	70.39	68.28	68.12	76.85	72.49
Actions	**YOWO + Multi-task precision**	**recall**	**F1**	**YOWO + FPN + Multi-task precision**	**recall**	**F1**
Raise hands	72.03	86.50	79.27	72.16	88.53	80.35
Adjust glasses	66.19	85.32	75.76	66.45	85.72	76.28
Touch forehead	66.08	70.39	68.28	68.13	72.26	70.20

It can clearly be seen from Table 3 that the application of the multi-scale and the multitask model improves the F1-score of raising hands and adjusting glasses by 10%. The improvement in the score is mainly due to the following reasons.

1. The use of FPN improves the feature extraction ability in a long-distance and complex background. FPN's application has increased the recall value by about 10%;
2. When we recognize the hand-raising movement, not only is the whole movement recognized, but also the raised brand. When we recognize the action of holding glasses, not only is the whole movement, but also the glasses. The use of the multi-task learning model greatly improves the accuracy of the model for the location. This increases the degree of improvement of the recall value up to about 16% and improves the value of precision, so the recognition effect of the model is enhanced.

It can be seen from the five datasets in Table 3 that the separate introduction of FPN or muliti-task has a positive effect on the experimental results. However, when these two techniques are superimposed, two of the three datasets obtain good results, but the result of the other set is not as good as when adding FPN separately. The possible reasons for the result are as follows: FPN and multi-tasks are used to enhance feature extraction. There are some interference features in these extracted features. These interference features have a negative impact on the model. Therefore, we will filter the extracted features and carry out further research on the optimization of feature extraction in the subsequent time.

We take the self-made dataset as an example. The original pictures, the accurate location block diagram, and the predicted block diagram are shown in Figures 10–12.

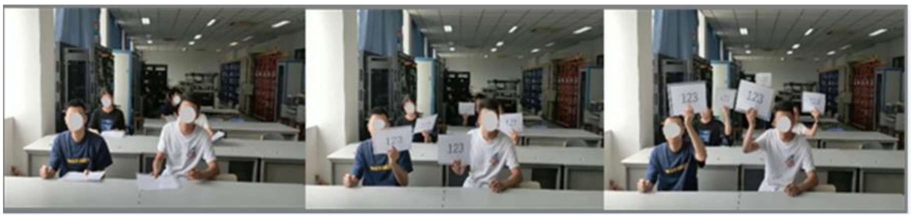


Figure 10. Original pictures of multi-player action.

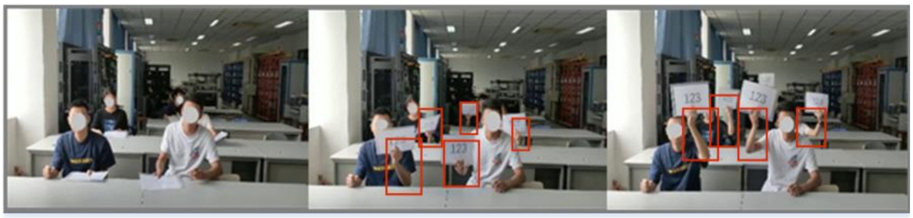


Figure 11. The accurate location of multiplayer action.

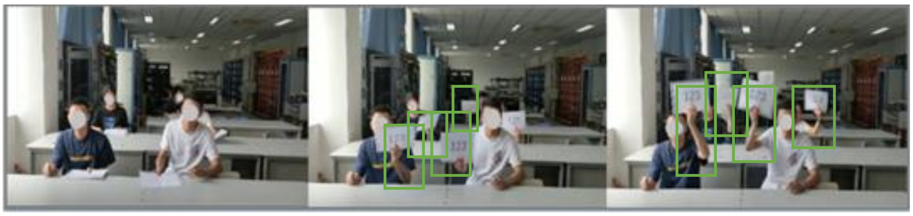


Figure 12. The predicted location of multiplayer action.

The introduction of the multi-scale feature extraction module and the multitask learning module increases the calculation and parameters of the model, so the separable convolution mode is introduced. The amount of computation, parameters, and training time under the two different modes are shown in Table 4.

Table 4. Comparison of performance efficiency under the different convolution modes.

Different Models	Calculation Amount (m)	Parameter Quantity (m)	Inference Time (s)
Ordinary convolution	445.96	0.669	0.005
Separable convolution	113.58	0.538	0.004

The use of separable depth convolution reduces the computation by 75% compared with the original model, the parameter quantity is reduced by 16.7%, and the inference time is reduced by 20% when we train it once with four batches. The efficiency of the model is accelerated. This is due to the method of depth separable convolution, which separates ordinary convolution into two parts: deep convolution and point-wise convolution. Deep convolution is responsible for filtering, acting on each input channel. Point-wise convolution is responsible for converting the channel and acting on the output feature map of deep convolution. They reduce the parameters and difficulty of convolution.

We not only optimize the classification of action but also improve the precision of the start-end frame location. The accuracy of temporal action detection on different actions is improved, and the improvement is more obvious when the duration of the action is shorter. The accuracy of different action is shown in Table 5.

Table 5. The accuracy of temporal detection on self-made dataset.

Action	YOWO			Our Method		
	IoU = 0.2	IoU = 0.3	IoU = 0.5	IoU = 0.2	IoU = 0.3	IoU = 0.5
Raise hands	72.6	70.2	56.5	75.2	74.1	67.3
Adjust glasses	75.3	73.3	59.0	77.6	76.2	70.2
Touch forehead	78.2	75.8	60.9	79.8	76.7	71.1

The performance of temporal detection's accuracy on the overall self-made dataset is shown in Figure 13. The accuracy of video-mAP increased from 58.1 to 69.2 when IoU was set as 0.5.

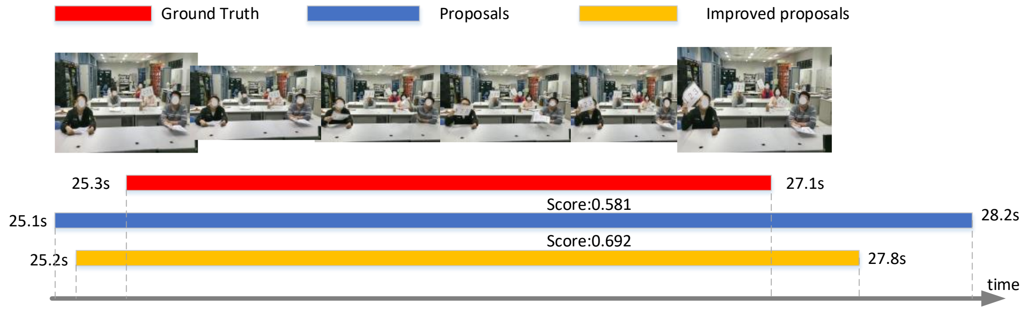


Figure 13. The accuracy of temporal action detection.

Temporal action detection is used to detect the first frame of action and the last frame of the action. In the experiment, each continuous sixteen frames are taken as a group, and we use one frame of the group as the keyframe to determine the action confidence score.Therefore, the first frame may be misjudged as the start-frame, while 16th frame is the accurate start-frame. The model is judged in groups, which leads to the problem of the preposition of the start-frame. Therefore, when the average confidence score in the group is used as the criterion of start-frame, the problem of the preposition of the starting frame is relieved.

At the end of the action, we kept raising a hand, which leads to the high confidence score of the model to the lifting action. The average score of the whole group of frames is also at a higher level. This leads to the postposition of the stop-frame. After the boundary gradient threshold is introduced, the problem of postposition is alleviated. When the confidence score is very high, but the gradient is slight, the current frame is judged as the stop-frame. Therefore, the problem of the postposition of the stop-frame is relieved.

5. Conclusions

In this paper, the ordinary camera is used as the data collector. We use convolutional networks to judge and recognize the continuous sixteen frames. The multi-scale feature extraction module is introduced to extract multi-level and multi-scale features. This way, we can increase the visual perception field and improve the accuracy of target recognition. Meanwhile, the secret relationship between multiple tasks is used to explore the correlation between multiple tasks, and the accuracy of the action recognition model is further improved. At the same time, the separable depth convolution module is added to reduce the number of the model's parameters. This improves the model's efficiacy. These changes have a positive effect on the optimization of the model.

In this paper, the time of the action process is determined by locating the boundary of the time sequence. The confidence gradient evaluation term is proposed to improve the accuracy of the time sequence boundary detection. As the optimization method is only suitable for a fast-transient action video, it has some limitations for the slow-changing action video. In the following research, we will focus on expanding the information sources of the model by introducing other information sources, such as auditory information. Then, we can conduct action research within the framework of audio-visual fusion to obtain a better performance. At the same time, we will work hard to integrate different networks. We will try to integrate CNN and LSTM. We hope to make the best use of their advantages and bypass their disadvantages. We hope that this has a good performance not only in terms of accuracy, but also effectiveness.

Author Contributions: Conceptualization, Y.X. and F.Z.; Data curation, K.Z. and W.P.; Formal analysis, L.W.; Investigation, K.Z.; Methodology, W.P.; Resources, Y.X. and F.Z.; Software, F.Z.; Supervision, L.W.; Validation, Y.X. and F.Z.; Visualization, K.Z. and L.W.; Writing—original draft, Y.X.; Writing—review and editing, F.Z. and L.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China (Grant No. 61973178) and the Smart Grid Joint Fund of State Key Program of National Natural Science Foundation of China (Grant No. U2066203).

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

Abbreviations

The following abbreviations are used in this manuscript:

2D-CNN	Two Dimensions Convolutional Neural Networks
3D-CNN	Three Dimensions Convolutional Neural Networks
CNN	Convolutional Neural Networks
RNN	Recurrent Neural Networks
LSTM	Long Short-Term Memory
FPN	Feature Pyramid Network
YOWO	You Only Watch Once
SGD	Stochastic Gradient Descent

References

1. Lamghari, S.; Bilodeau, G.-A.; Saunier, N. A Grid-based Representation for Human Action Recognition. In Proceedings of the 2020 25th International Conference on Pattern Recognition (ICPR), Milan, Italy, 10–15 January 2021.
2. Liciotti, D.; Bernardini, M.; Romeo, L.; Frontoni, E. A sequential deep learning application for recognising human activities in smart homes. *Neurocomputing* **2020**, *396*, 501–513. [CrossRef]
3. Liu, J.; Wang, G.; Hu, P.; Duan, L.Y.; Kot, A.C. Global context-aware attention lstm networks for 3d action recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 1647–1656.
4. Liu, J.; Shahroudy, A.; Xu, D.; Kot, A.C.; Wang, G. Skeleton-Based Action Recognition Using Spatio-Temporal LSTM Network with Trust Gates. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *40*, 3007–3021. [CrossRef] [PubMed]
5. Si, C.; Chen, W.; Wang, W.; Wang, L.; Tan, T. An Attention Enhanced Graph Convolutional LSTM Network for Skeleton-Based Action Recognition. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 16–20 June 2019; pp. 1227–1236.
6. Liu, J.; Wang, G.; Duan, L.Y.; Abdiyeva, K.; Kot, A.C. Skeleton-Based Human Action Recognition With Global Context-Aware Attention LSTM Networks. *IEEE Trans. Image Process.* **2018**, *27*, 1586–1599. [CrossRef] [PubMed]
7. Xue, W.; Zhao, H.; Zhang, L. Encoding Multi-resolution Two-Stream CNNs for Action Recognition. In *International Conference on Neural Information Processing*; Springer: Cham, Switzerland, 2016.
8. Antunes, J.; Abreu, P.; Bernardino, A.; Smailagic, A.; Siewiorek, D. Attention Filtering for Multi-person Spatiotemporal Action Detection on Deep Two-Stream CNN Architectures. *arXiv* **2019**, arXiv:1907.12919.
9. Tu, Z.; Xie, W.; Qin, Q.; Poppe, R.; Veltkamp, R.C.; Li, B.; Yuan, J. Multi-stream CNN: Learning representations based on human-related regions for action recognition. *Pattern Recognit.* **2018**, *79*, 32–43. [CrossRef]
10. Park, S.K.; Chung, J.H.; Kang, T.K.; Lim, M.T. Binary dense sift flow based two stream CNN for human action recognition. *Multimed. Tools Appl.* **2021**. [CrossRef]
11. Shou, Z.; Wang, D.; Chang, S.F. Temporal Action Localization in Untrimmed Videos via Multi-stage CNNs. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 27–30 June 2016.
12. Gao, J.; Yang, Z.; Chen, K.; Sun, C.; Nevatia, R. TURN TAP: Temporal Unit Regression Network for Temporal Action Proposals. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 3648–3656.
13. Xiong, Y.; Zhao, Y.; Wang, L.; Lin, D.; Tang, X. A Pursuit of Temporal Accuracy in General Activity Detection. *arXiv* **2017**, arXiv:1703.02716.
14. Tran, D.; Bourdev, L.; Fergus, R.; Torresani, L.; Paluri, M. Learning Spatiotemporal Features with 3D Convolutional Networks. In Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 11–18 December 2015; pp. 4489–4497.
15. Simonyan, K.; Zisserman, A. Two-Stream Convolutional Networks for Action Recognition in Videos. *arXiv* **2014**, arXiv:1406.2199.

16. Feichtenhofer, C.; Pinz, A.; Zisserman, A. Convolutional Two-Stream Network Fusion for Video Action Recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 1933–1941.
17. Shen, H.; Meng, X.; Zhang, L. An Integrated Framework for the Spatio–Temporal–Spectral Fusion of Remote Sensing Images. *IEEE Trans. Geosci. Remote Sens.* **2016**, *54*, 7135–7148. [CrossRef]
18. Zhou, T.; Li, J.; Wang, S.; Tao, R.; Shen, J. MATNet: Motion-Attentive Transition Network for Zero-Shot Video Object Segmentation. *IEEE Trans. Image Process.* **2020**, *29*, 8326–8338. [CrossRef] [PubMed]
19. Sun, X.; Zhu, Q.; Qin, Q. A Multi-Level Convolution Pyramid Semantic Fusion Framework for High-Resolution Remote Sensing Image Scene Classification and Annotation. *IEEE Access* **2021**, *9*, 18195–18208. [CrossRef]
20. Yu, J.; Zhu, C.; Zhang, J.; Huang, Q.; Tao, D. Spatial Pyramid-Enhanced NetVLAD With Weighted Triplet Loss for Place Recognition. *IEEE Trans. Neural Netw. Learn. Syst.* **2020**, *31*, 661–674. [CrossRef] [PubMed]
21. Guo, C.; Wang, H.; Jian, T.; He, Y.; Zhang, X. Radar Target Recognition Based on Feature Pyramid Fusion Lightweight CNN. *IEEE Access* **2019**, *7*, 51140–51149. [CrossRef]
22. Tu, Z.; Li, H.; Zhang, D.; Dauwels, J.; Li, B.; Yuan, J. Action-Stage Emphasized Spatiotemporal VLAD for Video Action Recognition. *IEEE Trans. Image Process.* **2019**, *28*, 2799–2812. [CrossRef] [PubMed]
23. Ouyang, X.; Xu, S.; Zhang, C.; Zhou, P.; Yang, Y.; Liu, G.; Li, X. A 3D-CNN and LSTM Based Multi-Task Learning Architecture for Action Recognition. *IEEE Access* **2019**, *7*, 40757–40770. [CrossRef]
24. Liu, A.A.; Su, Y.T.; Nie, W.Z.; Kankanhalli, M. Hierarchical Clustering Multi-Task Learning for Joint Human Action Grouping and Recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 102–114. [CrossRef] [PubMed]
25. Zhou, T.; Qi, S.; Wang, W.; Shen, J.; Zhu, S.C. Cascaded Parsing of Human-Object Interaction Recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**. [CrossRef]
26. Shou, Z.; Chan, J.; Zareian, A.; Miyazawa, K.; Chang, S.F. CDC: Convolutional-De-Convolutional Networks for Precise Temporal Action Localization in Untrimmed Videos. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 1417–1426.
27. Zhou, T.; Li, J.; Li, X.; Shao, L. Target-Aware Object Discovery and Association for Unsupervised Video Multi-Object Segmentation. In Proceedings of the 2020 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 16–18 June 2020.
28. Kopuklu, O.; Wei, X.; Rigoll, G. You Only Watch Once: A Unified CNN Architecture for Real-Time Spatiotemporal Ac-tion Localization. *arXiv* **2019**, arXiv: 1911.06644.
29. Lin, T.; Zhao, X.; Su, H.; Wang, C.; Yang, M. BSN: Boundary Sensitive Network for Temporal Action Proposal Generation. In Proceedings of the European Conference on Computer Vision, Munich, Germany, 8–14 September 2018.