

Received December 13, 2020, accepted February 1, 2021, date of publication February 12, 2021, date of current version July 5, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3058998

Complete Video-Level Representations for Action Recognition

MIN LI^{1,2,3}, RUWEN BAI^{2,3}, BO MENG⁴, JUNXING REN^{2,3}, MIAO JIANG^{2,3}, YANG YANG^{1,2,3}, LINGHAN LI^{2,3}, AND HONG DU¹

¹School of Computer and Information Technology, Beijing Jiaotong University, Beijing 100044, China

²Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, China

³School of Cyber Security, University of Chinese Academy of Sciences, Beijing 100049, China

⁴School of Optics and Electronics, Beijing Institute of Technology, Beijing 100081, China


Corresponding author: Min Li (limin@iie.ac.cn)

ABSTRACT In most of the existing work for activity recognition, 3D ConvNets show promising performance for learning spatiotemporal features of videos. However, most methods sample fixed-length frames from the original video, which are cropped to a fixed size and fed into the model for training. In this manner, two problems limit the model performance for recognition. First, the cropped video clips are incomplete or even distorted in appearance, resulting in a large gap between the feature representation and semantics of human activity. Second, the useful features of longer video frame sequences are weakened by the repeated stacking of 3D convolution over deep networks due to the limitations of GPU memory and computing ability. This article proposes a method based on a 3D backbone network for multi scale spatial feature representation, which uses a pyramid pooling layer to allow the input of video frames at different scales, and then aggregates short-term spatial-temporal features into a long-term video-level representation. Objection detection is used as a component of model testing to explore the improvement of activity recognition considering the large amount of space-time redundancy in real life videos. An experiment is performed on the principal video dataset, UCF101, and the proposed method presents a competitive performance.

INDEX TERMS 3D ConvNets, activity recognition, video-level feature representation.

I. INTRODUCTION

Analyzing and understanding human activity in videos can be widely used in real-life scenarios such as intelligent video surveillance, nursing home care, and smart retail, and self-driving and human-computer interaction can benefit from the task of activity recognition in videos. In recent years, deep neural networks for image tasks exhibit a superior performance, followed by great progress in the study of video behavior recognition. For 3D video data with spatial and temporal components, the performance of the model for activity recognition depends heavily on the ability to learn robust spatial-temporal features from the video and obtain spatial and temporal dependencies [1]. Existing large-scale video datasets are collected from real, unconstrained environments, where the spatially complex video content and temporally varying activity durations make activity recognition in videos challenging.

The associate editor coordinating the review of this manuscript and approving it for publication was Tossapon Boongoen .

In terms of spatial complexity, real-world videos have many activity-independent factors, including complex backgrounds, perspective changes, lighting changes, human-scale changes, and movement speed. In terms of temporal duration, humans perform actions ranging from 1 second to more than 10 seconds. Complex actions last longer and consist of simple actions, and discrimination based on short-range temporal information is likely to lead to false activity predictions, as shown in the top of Fig. 1.

To overcome the above obstacles, most research work on activity recognition relies on deep convolutional neural networks (CNNs) with the powerful ability to extract appearance features to obtain spatial information about activity in video frames, and temporally focus on capturing video-level representations over a long-term range. A typical two-stream architecture, TSN [2], uses sparse sampling to obtain segments from long video sequences and aggregates video-level expressions based on the consensus of all segments, where the optical flow that is computationally expensive represents interframe dynamic changes.

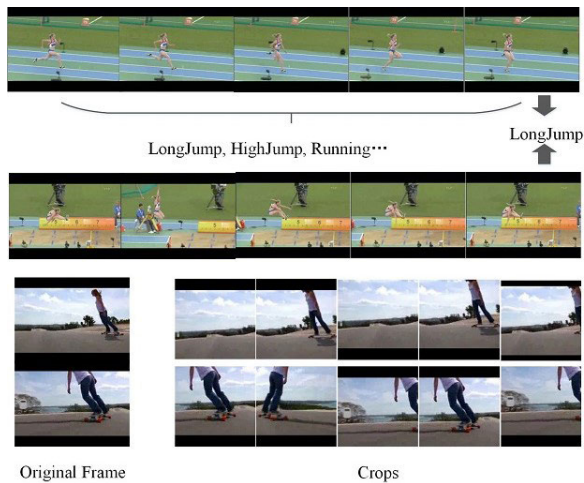


FIGURE 1. Short-term feature extraction and incomplete cropping leading to action misclassification.

3D convolutional networks (3D ConvNets) [3], [4] represent spatial-temporal features of video well, but their performance mostly rely on dense temporal sampling with predetermined sampling intervals and may incur excessive computational costs if used to process long videos. The duration of the videos fed into the model is limited by the available memory space. Long Short-Term Memory (LSTM) [5] performs sequence modeling well, but it has several shortcomings in expressing and aggregating spatial characteristics.

For deep learning methods, data substantially affect the performance of the model. Preprocessing before the model loads the videos can suppress unexpected noise in frames and highlight useful task-related information. In the above-mentioned research on activity recognition, generally, original videos are first converted into a fixed size, and video frames are sampled at a fixed interval to form fixed-length clips, which are randomly resized and cropped into a fixed size and then fed into the network. This processing can be regarded as data enhancement to prevent overfitting in model training, and the fixed crop size allows the output features of CNN to be flattened into a fixed dimension and then fed into fully connected layers. However, the incomplete context in cropped or distorted video clips also leads to incomplete or even distorted human activities recognition. The bottom of Fig. 1 shows that incomplete action feature extraction and cropped or distorted video context result in incorrect action classification.

In this article, spatial and temporal dimensions are considered and represent human activities to avoid omitting important information related to activities in the video. Spatially, to avoid the problem of incomplete action details caused by frame cropping, a pyramidal pooling layer is used to allow CNN to accept frames of varying sizes. Temporally, short-term spatial-temporal features are learned through 3D convolution, then the long-term temporal representation is constructed by aggregating multiple clip-level features at the end, similar to TSN. In this manner, our approach temporally

models the long-term dynamics of the video and spatially obtains complete contextual information about the video.

Real-world videos contain too much spatial-temporal redundant information, which makes the model predict activity correctly difficult. Handling long, untrimmed videos for activity recognition models is challenging when models are deployed in real-world scenarios. Training a model from scratch for activity detection to locate action boundaries in long videos takes a large amount of time and effort, and demands laboriously labeling videos. Integrating domain knowledge to simplify the optimization of deep models is a common strategy in deep learning. In this article, object detection is utilized to focus on the human action area in the frame and concatenate them into an action block that is fed into the model for action prediction, which greatly improves efficiency without loss of accuracy during model testing.

In summary, the main contributions in this article are as follows:

- A pyramid pooling layer captures fixed-size features from multiscale feature inputs, thus avoiding the absence of critical context due to frame cropping and resizing. Without too much extra computing overhead, more complete action details can be retained.
- In the test phase, a lightweight object detection framework is utilized to extract the human action block from the original video, which further spatially and temporally removes redundant information in the video that is not related to the activity, improving the prediction efficiency and accuracy of the model for activity recognition. An experiment is performed on a large-scale video dataset, UCF101, and an advanced accuracy is achieved.
- The proposed method uses 3D convolution and later fusion, which simultaneously models short-term spatial-temporal features and long-term temporal action features in videos, and finally extracts video-level feature representations.

II. RELATED WORK

SPATIOTEMPORAL FEATURE REPRESENTATION

CNNs have a powerful ability to learn the appearance of images. Benefiting from the great progress in CNNs for image tasks [6]–[9], Simonyan and Zisserman [6] and Feichtenhofer *et al.* [10] utilized a pretrained 2D CNN model on the large-scale image dataset ImageNet [11] to form a two-stream framework with spatial and temporal branches, which extract the appearance features and interframe dynamics. However, such models focus more on appearance and process up to ten video frames at a time. Moreover, the temporal motion information between frames is captured by a dense optical flow with a large computational overhead that limits the scalability of the deep model on large video datasets. Yue-Hei Ng *et al.* [12] proposed using LSTM [13] to aggregate frame-level features from the CNN output to model

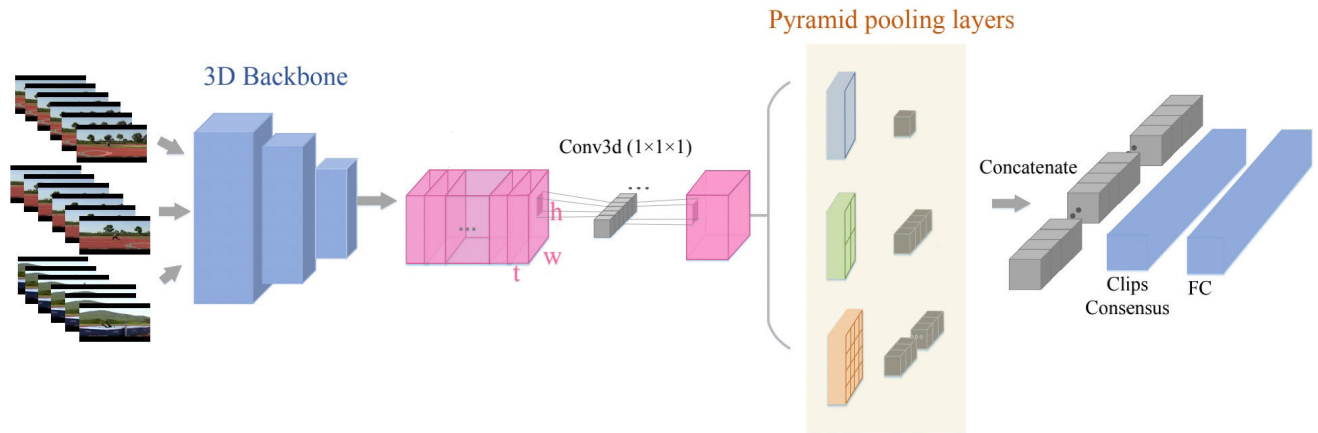


FIGURE 2. Our proposed 3D network architecture.

longer frame sequences, but it does not work well for videos due to the differences between video frames and speech and text.

Recently, more research work [3], [4], [14]–[16] focused on using 3D ConvNets for activity recognition in videos. Using 3D convolution to process 3D data like videos, which can simultaneously learn the spatial and temporal features of frame sequence, is intuitive. I3D [4] and P3D [14] benefited from loading the pretraining parameters on ImageNet, and Tran *et al.* [15] demonstrated the advantages of deep 3D ConvNets for learning video representations. Based on the research work P3D, R(2+1)D [17] adopted the (2+1)D decomposition spatiotemporal convolution on the ResNet3D backbone network. In this article, a deep 3D ConvNet is used to extract clip-level features in short frame sequences, and a later fusion strategy is finally adopted to aggregate short video clips into long-term and video-level representations.

MULTISCALE PYRAMID POOLING

The idea of pyramids is commonly used to solve multi-scale problems of detecting objects in images. For this problem, the construction of image pyramids [18] and feature pyramids [19] are traditional ways. The image pyramid structure [18] yields a series of image sequences of varying resolutions by resizing the original image with given scale factors. However, using CNN to extract features separately for each layer of image pyramids takes up a large amount of memory. Proper scales must be selected to generate image pyramids for different practical scenarios. Alternatively, feature pyramids [19] are constructed with features from different CNN layers, using a single-scale image as input, where top-down lateral connections between low-resolution high-level features and high-resolution low-level features allow features at all scales to represent rich semantic information.

In practice, the ability to represent features at different scales also differs, and up sampling makes the high-level semantics not always spread effectively. Both pyramid structures aim to obtain features at different scales in the image,

which retain detailed features but also add additional parameters and computational overhead. He *et al.* [20] proposed spatial pyramid pooling as a transition from the convolutional layer to the fully connected layer, allowing the convolutional network to receive different sizes of image inputs and retaining complete spatial information. Activity recognition in videos is more concerned with the foreground region of actions associated with people in the frame, whereas the position of a person in the current frame is variable, and any cropping may result in the loss of important cues associated with the action. Thus, preserving multiscale features is more beneficial to our task.

OBJECT DETECTION

People and objects interacting with people should be the focus of activity recognition. Zhang *et al.* [21] utilized only feature information related to people and objects by object detection. Similarly, Gao *et al.* [22] employed object detection methods to focus on human–object interaction in pictures. Two-stage RCNN [23], [24] and one-stage YOLO [25]–[28] object detection algorithms in deep learning show good performance. Overall, the two-stage object detection framework is more accurate, but the one-stage YOLO has faster, even real-time inference speeds with guaranteed accuracy, making it well suited for engineering practice. The latest YOLOv5 open-source project surpasses most of the object detectors in detection accuracy, while exceeding 300 fps detection speed. YOLOv5 is a flexible, lightweight network, which is extremely advantageous in the rapid deployment of the model. In this article, YOLOv5 [28] is used to preprocess videos during model testing. The preprocessed data are used to explore the contribution of object detection to activity recognition.

III. METHOD

The overall framework of our proposed method is shown in Fig. 2. For variable-length video input, the same sparse sampling strategy as TSN is used, dividing the whole input

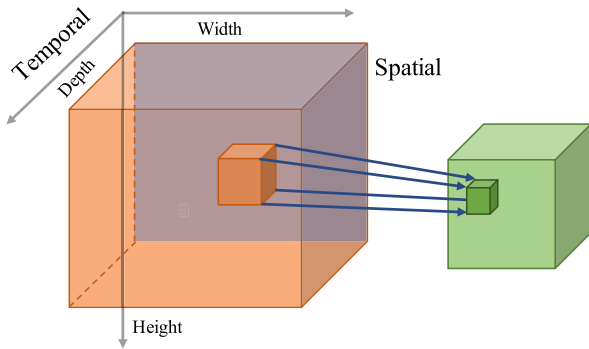


FIGURE 3. Convolution of 3D convolutional kernels.

video into K segments, and a few frames are sampled from K segments to extract video clips, which are fed into the 3D convolutional backbone network. A 3D pyramid pooling layer is used to aggregate the convolutional features of different spatial scales, form a unified feature map, preserve the complete action details, and achieve the transition from convolutional features of different sizes to the fully connected layer. A $1 \times 1 \times 1$ 3D convolution is used to control the output feature dimension of the 3D pyramid pooling layer and prevent generating an overly high feature dimension, which leads to a drastic increase of the full connection layer parameters. Finally, features from short video clips are aggregated into video-level feature representations.

3D SPATIAL-TEMPORAL MODELING

3D ConvNets provide a paradigm for efficiently learning the spatial features in videos and simultaneously attaining information about the temporal dynamics between frames. Convolution operations in 3D ConvNets extend 2D plane convolution operations over 3D space, as shown in Fig. 3. In terms of data input, the input into a 3D convolutional layer is a cube of multiple images along the time dimension, whereas the input to a 2D convolutional layer is a single image. The 3D convolutional layers are computed in the same manner as in 2D, with the cube-shaped convolutional kernels sliding over successive multiframe images. The output is still a 3D cube composed of multiple 2D maps. The resulting feature map not only reflects the relationships between pixels within a single image but also mines the correlation between contiguous frames in the time series, preserving the temporal information of the video.

As shown in Table 1, ResNet3D is adopted as the backbone network to extract 3D spatial-temporal features. Pretraining parameters loaded on ImageNet are always beneficial for 3D networks. Therefore, our work decomposes the 3D convolution into $(2 + 1)$ D convolution as in I3D, which reduces the number of parameters in the 3D convolution kernel, while being able to load pretrained models on ImageNet.

PYRAMID POOLING LAYER

In terms of data input, the input into a 3D convolutional layer is a cube of multiple images along the time dimension,

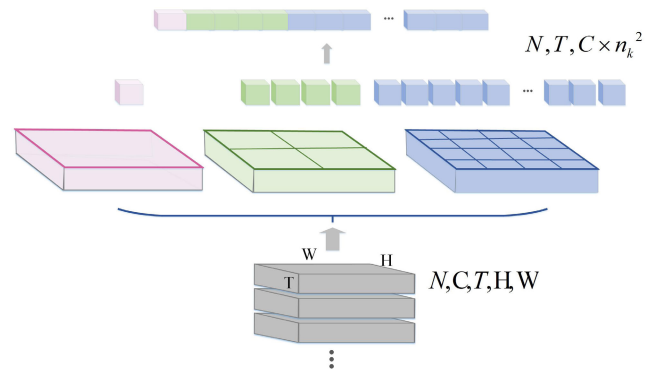


FIGURE 4. 3D pyramid pooling layer.

whereas the input to a 2D convolutional layer is a single image. The 3D convolutional layers are computed in the same manner as in 2D, with the cube-shaped convolutional kernels sliding over successive multiframe images. The output is still a 3D cube composed of multiple 2D maps. The resulting feature map not only reflects the relationships between pixels within a single image but also mines the correlation between contiguous frames in the time series, preserving the temporal information of the video.

ResNet3D is adopted as the backbone network to extract 3D spatial-temporal features. Pretraining parameters loaded on ImageNet are always beneficial for 3D networks. Therefore, our work decomposes the 3D convolution into $(2 + 1)$ D convolution as in I3D, which reduces the number of parameters in the 3D convolution kernel, while being able to load pretrained models on ImageNet.

Frame sequences with different crop sizes are fed into the 3D backbone, resulting in 3D feature maps with different sizes. The fully connected layer for classification in CNNs requires a fixed input feature size. The transition from the convolutional output features to the fully connected layer can be well achieved by using a pyramid pooling layer, which transforms features of different scales into fixed-size feature maps and extracts features at multiple levels and scales to improve the robustness of the model. The 3D pyramid pooling works are shown in Fig. 4.

Different scales of 3D pooling kernels are designed to unify the spatial dimension of the 3D convolutional feature map. For the pooling operation, the temporal dimension is kept unchanged, and the output dimension is unified only by changing the size of the pooling kernel in the spatial dimension. Different pooling scales are used to partition the feature map in the spatial dimension. Each level of the pyramid pooling layer corresponds to a pooling scale. For example, setting up three pooling scales means three levels of pyramid pooling layers. Fig. 4 shows three different scales (1×1 , 2×2 , 4×4) are used for partitioning. The input feature map size for the pyramid pooling layer is assumed to be $[N, C, T, H, W]$, where N is the batch size, C is the channels of the feature map, H, W, T are the height, width, and depth of the feature map respectively. K denote the number of

TABLE 1. 3D Backbone. Following [4], our 3D ResNet-50 backbone for extracting spatial-temporal features is shown. The output size and kernel size are in $T \times W \times H$ shape.

| Stage | Layer | Output size |
|-------------------|--|----------------------------|
| Raw video samples | — | $32 \times 340 \times 256$ |
| Input clips | — | $8 \times 340 \times 256$ |
| conv ₁ | $1 \times 7 \times 7, 64$, stride 1,2,2 | $8 \times 170 \times 128$ |
| pool ₁ | $1 \times 3 \times 3$, max, stride 1,2,2 | $4 \times 85 \times 64$ |
| res ₂ | $\begin{bmatrix} 1 \times 1 \times 1, 64 \\ 1 \times 3 \times 3, 64 \\ 1 \times 1 \times 1, 256 \end{bmatrix} \times 3$ | $4 \times 85 \times 64$ |
| res ₃ | $\begin{bmatrix} 1 \times 1 \times 1, 128 \\ 1 \times 3 \times 3, 128 \\ 1 \times 1 \times 1, 512 \end{bmatrix} \times 4$ | $4 \times 43 \times 32$ |
| res ₄ | $\begin{bmatrix} 3 \times 1 \times 1, 256 \\ 1 \times 3 \times 3, 256 \\ 1 \times 1 \times 1, 1024 \end{bmatrix} \times 6$ | $4 \times 22 \times 16$ |
| res ₅ | $\begin{bmatrix} 3 \times 1 \times 1, 512 \\ 1 \times 3 \times 3, 512 \\ 1 \times 1 \times 1, 2048 \end{bmatrix} \times 3$ | $4 \times 11 \times 8$ |

levels of the pyramid pooling layer, and each level contains a pooling scale n_k . Pooling operation concatenates the temporal feature map per layer by channel, and channel size at each level expands $n_k \times n_k$. After each pooling, the depth of the feature map temporally remains unchanged, which retains short-term temporal information. The pooling kernel in the spatial feature is calculated as in Eq. (1). The input of feature maps with multiscale sizes is unified into a fixed-size output by setting levels of pyramid pooling and the pooling scale at each level.

$$\begin{aligned}
 \text{kernel}(k_h, k_w) &= \text{ceil}\left(\frac{h}{n_k}, \frac{w}{n_k}\right) \\
 \text{stride}(s_h, s_w) &= \text{floor}\left(\frac{h}{n_k}, \frac{w}{n_k}\right) \\
 \text{padding}(p_h, p_w) &= \text{floor}\left(\frac{k_h \times n_k - h + 1}{2}, \frac{k_w \times n_k - w + 1}{2}\right)
 \end{aligned} \quad (1)$$

Pooling changes the size of the input feature map, not the dimensions, that is, the number of feature map channels. Directly feeding the output feature map of a deep 3D ConvNets into the pyramid pooling will result in a dramatic increase in the dimensionality of the output feature map, as well as excessive parameters in the fully connected layer. In this article, 3D convolution with a kernel size of $1 \times 1 \times 1$ is used to reduce the feature dimension.

VIDEO-LEVEL FEATURE REPRESENTATION

Repeated and stacked 3D convolution makes the useful features of longer-range frames weakened and difficult

to capture. Therefore, the clip-level features must be aggregated into long-term video-level representations. Assuming that the whole video is divided into K segments, average pooling is used to aggregate the short-term features of K segments. In this article, the whole video is divided into four segments with fixed sampling intervals of two and eight frames for each segment, considering the memory and computation capacity of the GPU. The intuition behind average pooling is to utilize the average activation of all clips for activity recognition.

REMOVE SPATIAL-TEMPORAL REDUNDANCY

For noisy videos with complex backgrounds, several redundancies are observed in the single-frame images spatially and frame sequences temporally, which are not related to human actions but affect the final recognition performance. In this article, YOLOv5 is employed to preprocess videos during model testing. Fig. 5 shows that the human detection box is expanded by two times to form a human action area, which does not exceed the image boundary. The human action area on consecutive frames constitutes the action block fed into the model. Adopting such a strategy can further improve the accuracy and efficiency of activity recognition.

IV. EXPERIMENTS

In the section, the evaluation dataset and implementation details of the proposed method are first introduced. Then, good practices for 3D pyramid networks and long-term modeling are explored and compared with advanced methods. Finally, the application of the lightweight object detection framework YOLOv5 to optimize model performance for activity recognition is discussed.

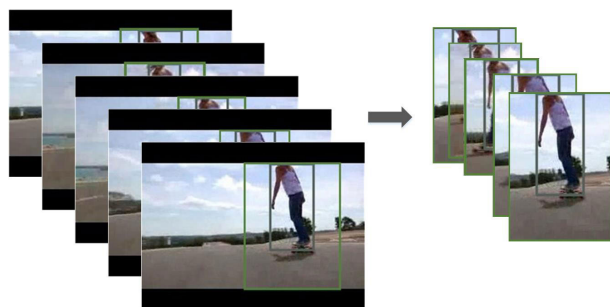


FIGURE 5. Yolo remove spatial-temporal redundancy.

DATASET

The UCF101 [29] dataset contains 5 major categories of everyday human activities: single-actor, instrument playing, character interaction, human interaction, and sports. The dataset is a collection of 13320 real-world videos from YouTube, with a total of 101 activity classes, each with 25 different actors, mostly adults. The duration of the videos vary depending on the complexity of the different actions. Most of the videos are recorded and uploaded by the users themselves, from unconstrained real-world environments, with large variations in cluttered backgrounds and resolution inconsistencies. Thus, the UCF101 dataset is very challenging and is a classic benchmark dataset for evaluating models for activity recognition. The UCF101 dataset contains three training/test splits, and the videos for training and testing in each subset are trimmed from different long videos. Over 3 split sets are evaluated.

IMPLEMENTATION DETAILS

ResNet 3D ConvNet is used as the backbone network. Except for the last fully connected layer for classification, each convolutional layer is followed by a batch normalization layer and a ReLU activation function. The parameters pre-trained on ImageNet are loaded during training. The Sports-1M dataset [30] is used for pretraining the 3D pyramidal pooling network and then trained on the UCF101 dataset using 6 NVIDIA Titan RTX GPUs. Batch size is 32, and initial learning rate is 1e-4, which could be adjusted automatically. The stochastic gradient descent algorithm optimizer is used, where momentum is set to 0.9. Before training, the video is cut into video frames at the frame rate of the original video. During training, considering the influence of GPU memory and computation, all video frames of a single video are divided into segments, and for each segment, the start is the earliest possible starting frame, and eight consecutive frames are selected to form each clip at a frame interval of 2. The spatial size of the shorter side of the video frame is adjusted to 256 pixels, and the resolution is kept the same.

ABLATION STUDY

A. 3D PYRAMID POOLING LAYER SETTINGS

In our 3D pyramid network, the levels of the pyramid pooling layer affect the model performance and the number

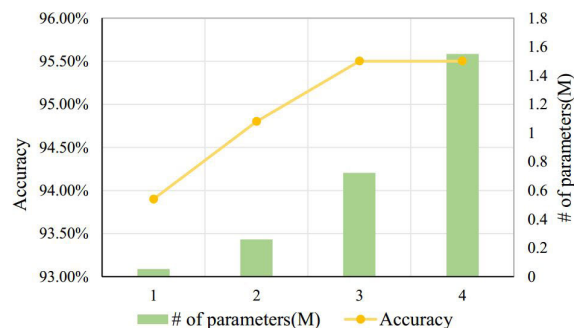


FIGURE 6. Accuracy and model parameters for different pyramid level settings on UCF101.

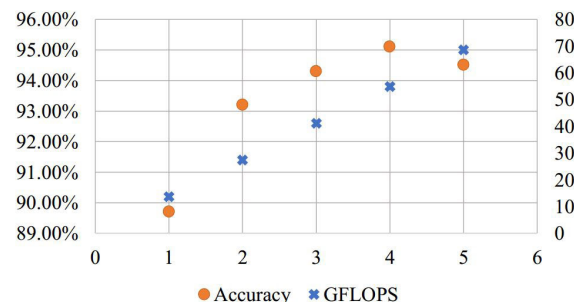


FIGURE 7. The effects of a different number of segments on computational overhead and model performance.

of parameters. Setting more levels of kernel scale leads to increasing channels of features, which are input into the fully connected layer. Accuracy is reported by averaging over all three splits. For example, if four pooling kernel scales are set up, the number of channels of the concatenated feature map is more than twice that of three scales. Fewer pyramid levels are selected while ensuring high accuracy to achieve a tradeoff between number of model parameters and performance. Fig. 6 shows the effect of various levels of the pyramid pooling layer on the number of parameters in the fully connected layer and the performance of the corresponding model on UCF-101. The three-level pyramid pooling layer is the optimal choice.

B. CONSIDER LONG-TERM INFORMATION

Tran et al. [15] found that clip-level test accuracy peaks when input reaches 32 frames. This finding implied that 3D ConvNets in practice obtain only a limited length of temporal information at a time. Thus, we argue that combining short-term temporal information from different parts of a long video is preferable to represent temporally complete activity information. An experiment is conducted on the UCF101 split1 test set with a different number of video segments and different lengths of a clip. The length of each clip is fixed to eight frames, and the effects of a different number of segments on computational overhead and model performance are compared, as shown in Fig. 7. If more frames are fed to the 3D network at once, the computational load on the GPU becomes difficult. When the number of segments is five, the

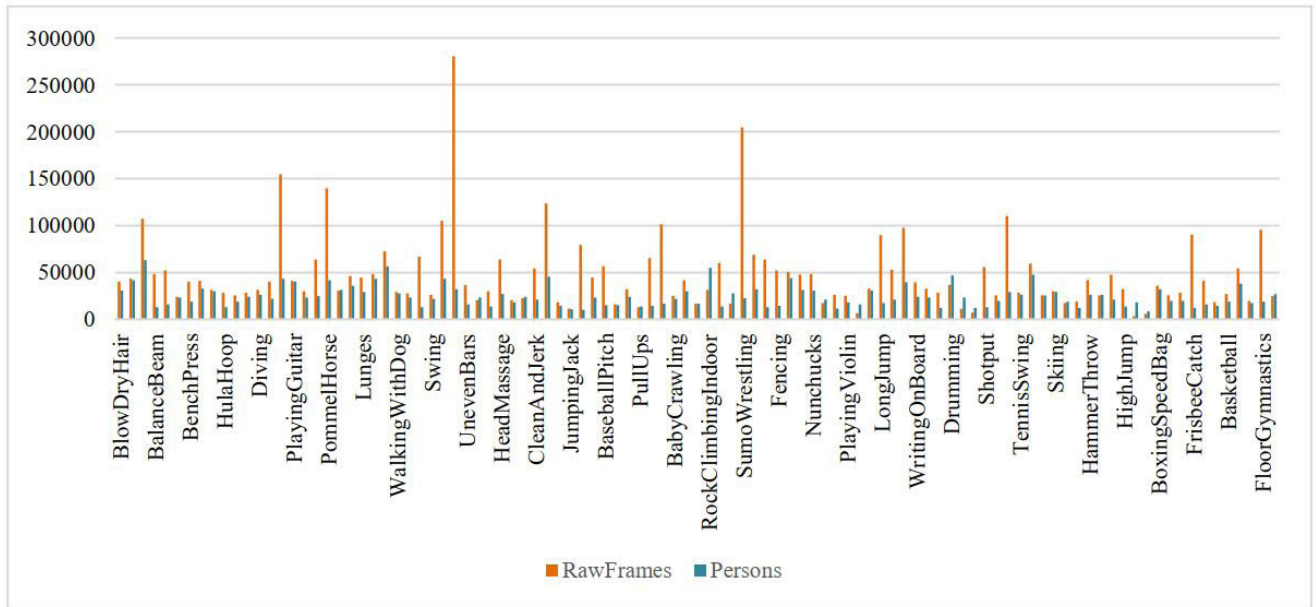


FIGURE 8. Useful data pre-processed with yolov5 compared to previous data.

TABLE 2. Accuracy improvements from long-term modeling.

| Frames per clip/segments | GFLOPS | Accuracy |
|--------------------------|--------|----------|
| 8/4 | 54.82 | 95.1% |
| 16/2 | 54.82 | 94.3% |
| 32/1 | 54.82 | 93.5% |

performance of the model decreases instead probably because of too much spacetime redundancy information.

Table 2 presents the test results after training on UCF101 split1 using clips consisting of different numbers of frames, and the computational overhead of the model is held constant. We can infer that considering the complete video temporal information is advantageous for recognizing activity without increasing the model computation.

C. COMPLEMENTARY TESTING FOR YOLOV5

YOLOv5 is used to preprocess videos in the UCF101 datasets. Fig. 8 compares the number of frames for each action-related category after preprocessing with the total number of frames in the original video. The preprocessed data eliminate numerous irrelevant spatial-temporal redundant information in each category, making the data well distributed. Our model is tested again on the preprocessed UCF101 data. The average test results of the three split sets are shown in Table 3. Object detection deployed in activity recognition provides performance improvements and can be used as a strategy to industrial applications. The results on the split1 test set of the preprocessed UCF101 dataset are visualized. Fig. 9 shows the confusion matrix. The data on the diagonal of the confusion matrix indicate that the prediction

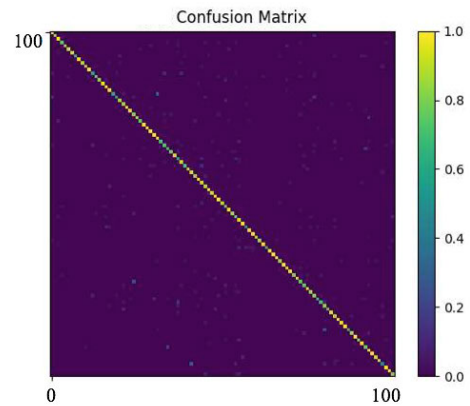


FIGURE 9. Confusion matrix on preprocessed UCF101.

TABLE 3. Test results on pre-processed UCF101.

| Methods | Accuracy |
|-------------------|----------|
| Ours model | 95.5% |
| YOLOv5+ Our model | 96.1% |

labels of the actions are the same as the true labels. Statistically, 41 of the 101 action categories are predicted with 100% accuracy.

COMPARISON WITH STATE-OF-THE-ART METHODS

Table 4 compares our method with state-of-the-art methods on the UCF101 dataset, using the average accuracy over three splits. Our method performs well when the model is trained on RGB data only, which is substantially better than

TABLE 4. Comparison with state of the art methods.

| Method | Pretraining dataset | Accuracy (%) |
|--------------------|---------------------|--------------|
| Two-stream | ImageNet | 88 |
| TSN-RGB | ImageNet | 86.5 |
| TSN-RGB+Flow(3seg) | ImageNet | 94.6 |
| TSN-RGB+Flow(7seg) | ImageNet | 94.9 |
| C3D | Sports1M | 85.2 |
| P3D | ImageNet+Sports1M | 88.6 |
| I3D-RGB+Flow | ImageNet | 93.4 |
| R(2+1)D-RGB | Sports1M | 93.6 |
| Ours | ImageNet+Sports1M | 95.5 |

classical two-stream and 3D ConvNets C3D. In the case of using the same pretraining datasets, our method achieves a higher accuracy by using a much smaller model than the 199-layer depth model P3D. Our method also outperforms R(2+1)D by 1.9%, which also demonstrates the advantage of loading the pretraining parameters of ImageNet. Compared with the two-stream model trained with optical flow, our method pretrained on Sports1M outperforms TSN 0.6% and I3D by 2.1%, which is pretrained on ImageNet. Our model shows a competitive performance.

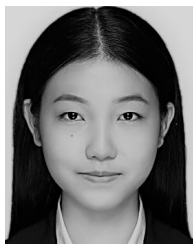
V. CONCLUSION

Previous methods cropped and distorted raw video data such that the network could only see incomplete action details at once. A 3D pyramid pooling network is proposed to extract complete video-level features. The proposed method uses a pyramid pooling layer to uniform feature maps of different scales as fixed sizes and finally aggregates clip-level features into video-level features to obtain more complete action features. The level setup of the pyramid pooling layer and the details of video-level feature aggregation are experimentally explored, demonstrating the robustness of multiscale spatial feature expression and the benefits of representing complete video-level action features. Our approach is competitive with advanced methods. The promising performance on the UCF101 dataset demonstrates the effectiveness of our method. In addition, the auxiliary role of the object detection framework for the activity recognition task is explored. However, we need to explore how to implement a more lightweight network for quickly identifying activities in a video in the future. We will also focus further on the detailed study of activity recognition models for industrial applications.

REFERENCES

- [1] H.-B. Zhang, Y.-X. Zhang, B. Zhong, Q. Lei, L. Yang, J.-X. Du, and D.-S. Chen, "A comprehensive survey of vision-based human action recognition methods," *Sensors*, vol. 19, no. 5, p. 1005, Feb. 2019.
- [2] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. Van Gool, "Temporal segment networks: Towards good practices for deep action recognition," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2016, pp. 20–36.
- [3] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3D convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 4489–4497.
- [4] J. Carreira and A. Zisserman, "Quo vadis, action recognition? A new model and the kinetics dataset," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6299–6308.
- [5] J. Donahue, L. A. Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, T. Darrell, and K. Saenko, "Long-term recurrent convolutional networks for visual recognition and description," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 2625–2634.
- [6] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 568–576.
- [7] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi, "Inception-v4, inception-ResNet and the impact of residual connections on learning," 2016, *arXiv:1602.07261*. [Online]. Available: <http://arxiv.org/abs/1602.07261>
- [8] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
- [9] D. Sun, Y. Yang, M. Li, J. Yang, B. Meng, R. Bai, L. Li, and J. Ren, "A scale balanced loss for bounding box regression," *IEEE Access*, vol. 8, pp. 108438–108448, 2020.
- [10] C. Feichtenhofer, A. Pinz, and A. Zisserman, "Convolutional two-stream network fusion for video action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 1933–1941.
- [11] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, May 2017.
- [12] J. Yue-Hei Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and G. Toderici, "Beyond short snippets: Deep networks for video classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 4694–4702.
- [13] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [14] Z. Qiu, T. Yao, and T. Mei, "Learning spatio-temporal representation with pseudo-3D residual networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 5533–5541.
- [15] D. Tran, J. Ray, Z. Shou, S.-F. Chang, and M. Paluri, "ConvNet architecture search for spatiotemporal feature learning," 2017, *arXiv:1708.05038*. [Online]. Available: <http://arxiv.org/abs/1708.05038>
- [16] M. Li, Y. Qi, J. Yang, Y. Zhang, J. Ren, and H. Du, "3D convolutional two-stream network for action recognition in videos," in *Proc. IEEE 31st Int. Conf. Tools with Artif. Intell. (ICTAI)*, Nov. 2019, pp. 1697–1701.
- [17] D. Tran, H. Wang, L. Torresani, J. Ray, Y. LeCun, and M. Paluri, "A closer look at spatiotemporal convolutions for action recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 6450–6459.
- [18] E. H. Adelson, C. H. Anderson, J. R. Bergen, P. J. Burt, and J. M. Ogden, "Pyramid methods in image processing," *RCA Eng.*, vol. 29, no. 6, pp. 33–41, 1984.
- [19] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2117–2125.
- [20] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 9, pp. 1904–1916, Sep. 2015.
- [21] Y. Zhang, P. Tokmakov, M. Hertz, and C. Schmid, "A structured model for action detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 9975–9984.
- [22] C. Gao, Y. Zou, and J.-B. Huang, "ICAN: Instance-centric attention network for human-object interaction detection," 2018, *arXiv:1808.10437*. [Online]. Available: <http://arxiv.org/abs/1808.10437>
- [23] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1440–1448.
- [24] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 91–99.
- [25] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 779–788.
- [26] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," 2018, *arXiv:1804.02767*. [Online]. Available: <http://arxiv.org/abs/1804.02767>

- [27] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal speed and accuracy of object detection," 2020, *arXiv:2004.10934*. [Online]. Available: <http://arxiv.org/abs/2004.10934>
- [28] G. Jocher, A. Stoken, J. Borovec, and P. Rai. *Ultralytics/Yolov5: V3.1—Bug Fixes and Performance Improvements*. Accessed: Oct. 2020. [Online]. Available: <https://doi.org/10.5281/zenodo.4154370>
- [29] K. Soomro, A. R. Zamir, and M. Shah, "UCF101: A dataset of 101 human actions classes from videos in the wild," 2012, *arXiv:1212.0402*. [Online]. Available: <http://arxiv.org/abs/1212.0402>
- [30] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthakar, and L. Fei-Fei, "Large-scale video classification with convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 1725–1732.



MIAO JIANG received the B.S. degree in computer science and technology from the Nanjing University of Science and Technology, China, in 2019. She is currently pursuing the M.S. degree in image processing technology based on deep learning with the Institute of Information Engineering, Chinese Academy of Sciences. Her research interests include object detection and action recognition.



MIN LI is currently a Research Professor with the Institute of Information Engineering, Chinese Academy of Sciences. He is also a Professor with the School of Cyber Security, University of Chinese Academy of Sciences. His research interests include computer vision and intelligent analysis, indoor target localization, and the Internet of Things.



RUWEN BAI received the B.S. degree from the School of Electronic Information Engineering, Hebei University, Hebei, China, in 2018. She is currently pursuing the Ph.D. degree with the Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China. Her current research interests include computer vision, video content analysis, and human activity recognition.



BO MENG received the M.S. degree in automation engineering from the Beijing Institute of Technology in 2007. He is currently pursuing the Ph.D. degree with the School of Optics and Photonics, Beijing Institute of Technology, China. From 2013 to 2014, he was a Visiting Scholar with the Department of Biomedical Engineering, Rensselaer Polytechnic Institute, Troy, USA. His research interests include computed tomography, spectral CT Imaging, and deep learning.



JUNXING REN received the M.S. degree in communication engineering from Beihang University, China, in 2014. She is currently an Intermediate Engineer with the Institute of Information Engineering, Chinese Academy of Sciences. Her research interests focus on information security, electromagnetic safety, and deep learning.



YANG YANG received the B.S. degree in computer science and technology from Beijing Jiaotong University, China, in 2017. He is currently pursuing the Ph.D. degree in image processing technology based on deep learning with the Institute of Information Engineering, Chinese Academy of Sciences. His research interests include object detection, action recognition, and pose estimation.



LINGHAN LI received the B.S. degree in communication engineering from Xidian University, China, in 2018. He is currently pursuing the master's degree in cyberspace security with the Institute of Information Engineering, Chinese Academy of Sciences. His current research interest is intelligent video analysis based on deep learning.



HONG DU is currently a Research Fellow and a Visiting Professor with Beijing Jiaotong University. His main research interests are cyberspace security and artificial intelligence.

...