



Article

Action Recognition Network Using Stacked Short-Term Deep Features and Bidirectional Moving Average

Jinsol Ha , Joongchol Shin, Hasil Park and Joonki Paik * 

Department of Image, Chung-Ang University, Seoul 06974, Korea; jinsol_ha@ipis.cau.ac.kr (J.H.); jcshin@ipis.cau.ac.kr (J.S.); hspark@ipis.cau.ac.kr (H.P.)

* Correspondence: paikj@cau.ac.kr

Abstract: Action recognition requires the accurate analysis of action elements in the form of a video clip and a properly ordered sequence of the elements. To solve the two sub-problems, it is necessary to learn both spatio-temporal information and the temporal relationship between different action elements. Existing convolutional neural network (CNN)-based action recognition methods have focused on learning only spatial or temporal information without considering the temporal relation between action elements. In this paper, we create short-term pixel-difference images from the input video, and take the difference images as an input to a bidirectional exponential moving average sub-network to analyze the action elements and their temporal relations. The proposed method consists of: (i) generation of RGB and differential images, (ii) extraction of deep feature maps using an image classification sub-network, (iii) weight assignment to extracted feature maps using a bidirectional, exponential, moving average sub-network, and (iv) late fusion with a three-dimensional convolutional (C3D) sub-network to improve the accuracy of action recognition. Experimental results show that the proposed method achieves a higher performance level than existing baseline methods. In addition, the proposed action recognition network takes only 0.075 seconds per action class, which guarantees various high-speed or real-time applications, such as abnormal action classification, human-computer interaction, and intelligent visual surveillance.

Keywords: action recognition; three-dimensional convolution (C3D); short-term pixel-difference; bidirectional moving average



Citation: Ha, J.; Shin, J.; Park, H.; Paik, J. Action Recognition Network Using Stacked Short-Term Deep Features and Bidirectional Moving Average. *Appl. Sci.* **2021**, *11*, 5563. <https://doi.org/10.3390/app11125563>

Academic Editors: Andrea Prati and Antonio Fernández-Caballero

Received: 20 March 2021

Accepted: 8 June 2021

Published: 16 June 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Action recognition is an important task in various video analytic fields, such as real-time intelligent monitoring, human-computer interaction, and autonomous driving systems [1–3]. Due to the nature of video processing, a high-speed or real-time processing is an essential condition of action recognition. To recognize an action, machine learning or deep-learning-based classifiers generally use motion features in a video sequence. Since the actions may or may not be continued for the entire frame, the video is treated as a frame acquired from an image sequence.

Since action recognition can be considered as an extended classification task for a set of multiple, temporally related image frames, many image-classification-based approaches were proposed using a convolutional neural network (CNN). The two-dimensional (2D) convolution neural net using 2D features can effectively recognize characteristics of an object. Alexnet is the first, simplest image classification network, which consists of eight 2D convolutional layers, max-pooling layers, dropout layers, and fully connected layers [4]. Since Alexnet, various CNN-based models with deeper layers were proposed to improve the classification performance. VGG16 consists of 16 convolutional layers of 3×3 filters, pooling layers, and fully-connected layers [5]. Similarly, various high-performance CNN-based classification models were proposed, such as Googlenet and Densenet [6,7]. However, since 2D CNN features learn only spatial context, there is a limit to the inclusion of temporal

features in a systematic way. Furthermore, it is difficult to recognize an action in a video frame composed of multiple images using a single image classification algorithm.

To process a video clip with multiple frames, Karpathy et al. used a 2D CNN structure to train 2D features for action recognition, and fused a part of the architecture to use additional information [8]. However, Karpathy's model can train only spatial features, and temporal features are applied to the final classification layer. If the frames are individually processed in the network, the temporal relation between each frame cannot be fully incorporated.

A 3D convolutional neural net (C3D) was proposed to overcome the structural limitation of the 2D CNN for action recognition. Since input, convolution filters, and feature maps in the C3D are all three-dimensional, it can learn both spatial and temporal information using a relatively small computation amount. Ji et al. extracted features using 3D convolution, and obtained motion information included in temporarily adjacent frames for action recognition [9]. Tran et al. proposed a C3D that reduces the number of parameters while maintaining the minimum kernel size [10]. Tran's C3D takes 16 frames as an input, which is not sufficient to analyze a long-term video. Although existing C3D-based approaches include some spatial–temporal information, they cannot completely represent the temporal relationship between each frame. To incorporate the inter-frame relationship, both spatial and temporal information need to be taken into account at the same time. In this context, optical-flow-based methods estimate motion between two temporally adjacent pixels in video frames. Lucas and Kanade used optical flow to recognize actions [11]. Specifically, they set up a window for each pixel in a frame, and matched the window in the next frame. However, this method is not suitable for real-time action recognition since pixel-wise computation requires a very large amount of computation.

To solve these problems, this paper presents a novel deep learning model that can estimate spatio-temporal features in the input video. The proposed model can recognize actions by combining spatial information and adjacent short-term pixel difference information. Since a 2D CNN module is used to estimate deep spatial features, the proposed method can recognize actions in real-time. In addition, by fusing the proposed model to the C3D, we can improve the recognition accuracy by analyzing the temporal relationship between frames. Major contributions of the proposed work include:

- By creating a deep feature map using a differential image, the network can analyze the temporal relationship. The proposed deep feature map can recognize actions using 2D CNN with temporal features;
- Since a human action is a series of action elements, it can be considered as a sequence. In this context, we propose a deep learning model to increase the accuracy of action recognition by assigning a high weight to an important action;
- Using the late fusion with C3D, we can improve the recognition accuracy, and prove that the temporal relation is important.

The paper is organized as follows: Section 2 describes related works for action recognition. Section 3 defines the short-term pixel-difference image, and presents the moving average network for action recognition. After Section 4 presents experimental results, Section 5 concludes the paper.

2. Related Works

In the action recognition research field, various methods are proposed to represent the temporal information of a video. Prior to the deep learning approach, the feature-engineering-based methods were the mainstream way to express the movement of an object in a video frame using hand-crafted features. A histogram of oriented optical flow (HOOF) was used to express motions [12]. HOOF represents the direction and speed of an actual flow as a vector. This is an operation between each pixel in temporally adjacent frames, and represents the optical flow of a moving object, excluding the stationary region in the video. The optical flow is considered as a histogram for each action and is classified as a corresponding action histogram.

Recently, various deep learning approaches were proposed for action recognition using CNNs. The proposed network uses VGG16 as a baseline model for image classification, as shown in Figure 1. All layers in VGG16 consist of 4096 3×3 convolutional layers, which increases the nonlinearity and accuracy of action recognition.

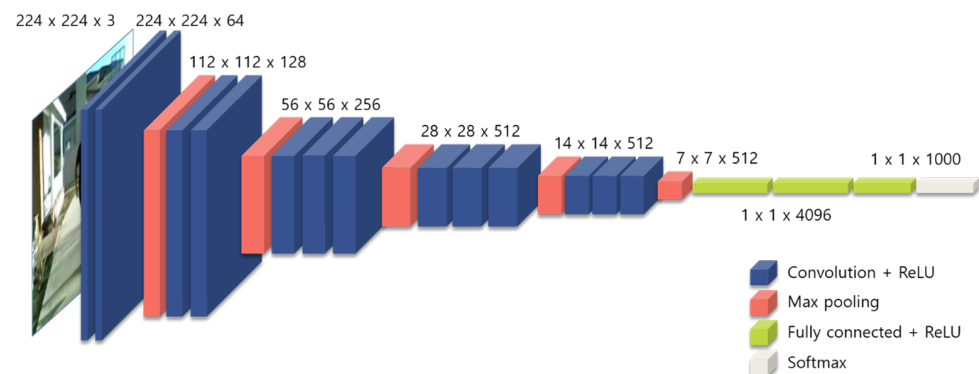


Figure 1. VGG16 network architecture.

Recognition of a video requires more computation than that of a single image, since it requires temporal features as well as spatial features. In this context, various studies are being conducted to process the relationship between spatial and temporal information. A 3D convolutional neural net (C3D) was proposed to train the temporal features of input video. The C3D uses both temporal and spatial features from input video. Features in the C3D layers create blocks including both spatial and temporal coordinates. The C3D block promotes spatial–temporal learning for part of the consecutive frames in the convolution features. Figure 2 shows the structure of 2D and 3D convolution features.

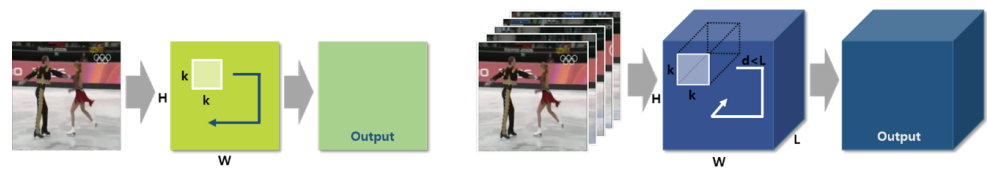


Figure 2. Structure of 2D and 3D convolution features.

3. Proposed Action Recognition Network

In this section, we present a novel action recognition network (ARN) using deep feature maps and a bidirectional exponential moving average, as shown in Figure 3. ARN consists of four sequential steps: (i) video frame sampling, (ii) the generation of deep feature maps that include temporal features, (iii) bidirectional exponential moving average network that assigns a high weight to an important action, and (iv) calculation of the action class loss and classification using late fusion with the C3D. The aforementioned four steps are described in the following subsections, respectively.

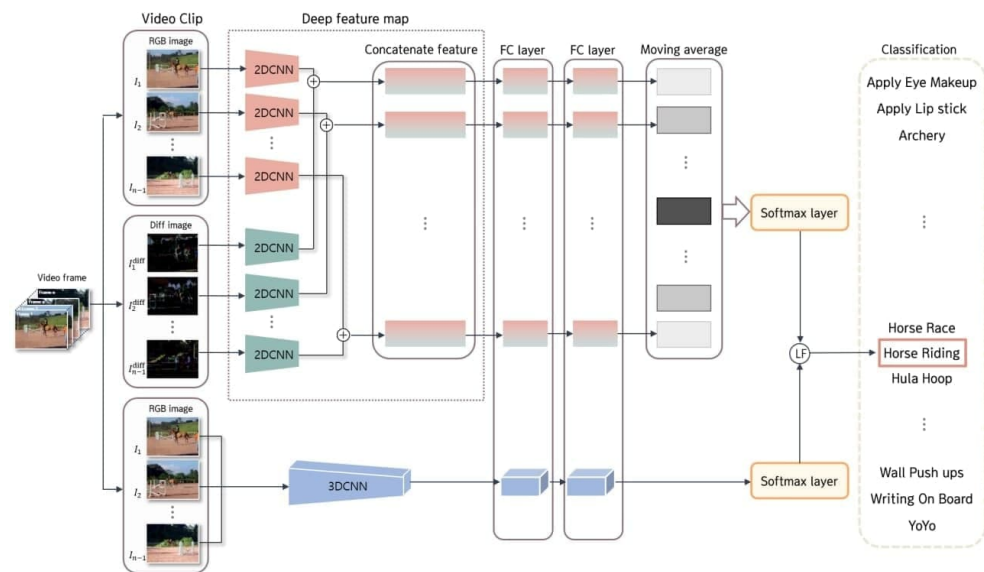


Figure 3. Architecture of the proposed action recognition network.

3.1. Video Frame Sampling

To implement the proposed method in an efficient manner, we sample video frames, where a single action continues for ten seconds, for example, 300 frames are generated assuming the frame rate 30 frame per second (fps). Due to the temporal redundancy between video frames, we need to sample the video to make a shorter video clip that preserves the action information.

Let $\{I_0^*, \dots, I_{T-1}^*\}$ represent a set of the initial video frames, then the set of sampled video frames is expressed as $\{I_0, \dots, I_{N-1}\}$, for $m = \lfloor T/N \rfloor$, where

$$I_n = \begin{cases} I_{mn}^*, & n = 0, \dots, N - 2 \\ I_{T-1}^*, & n = N - 1 \end{cases} \quad (1)$$

The sampled video frames have a uniform interval, and provide an ordered continuity between frames. In this work, we used $N = 16$ and resized each frame to 112×112 , based on the experimental best performance and computational efficiency.

3.2. Generation of Deep Feature Maps

In this subsection, we present a novel method to generate deep feature maps for accurate, efficient action recognition. Since action recognition requires both spatial and temporal information, we combined two types of feature maps of RGB and differential images. In addition, the generated feature maps not only include spatio-temporal information but also temporally continuous information between adjacent frames using temporal relations.

The RGB image feature map is generated to use the spatial information of the input video clip, which is expressed as

$$F_{RGB} = \phi(I_n(x, y)) \quad (2)$$

where I_n represents the n -th sampled video frame, (x, y) the pixel coordinate, and $\phi(\cdot)$ the bottleneck feature map of the VGG16 backbone network.

The differential feature map is generated using RGB frame with spatial information. Specifically, the RGB space of the sampled frames contains spatial information, and each sequence of frames contains temporal information. We generate a deep feature map using the temporal relationship between temporally adjacent frames by calculating pixel-wise difference between adjacent frames

$$I_n^{diff}(x, y) = I_n(x, y) - I_{n-1}(x, y), \text{ for } n = 1, \dots, N - 1 \quad (3)$$

Figure 4 shows the results of differential images from a set of selected UCF101 dataset [13]. A differential image does not have a still background, but moving objects. Figure 4 is a class containing human hands. When we generate a differential image from a video taken by a stationary camera, only the moving hand region mixing with the upper body of a person is detected. In the case of Figure 4a,f, some background regions remain due to the movement of the camera, but the pixel value of a person with major movements can be differentiated from background.

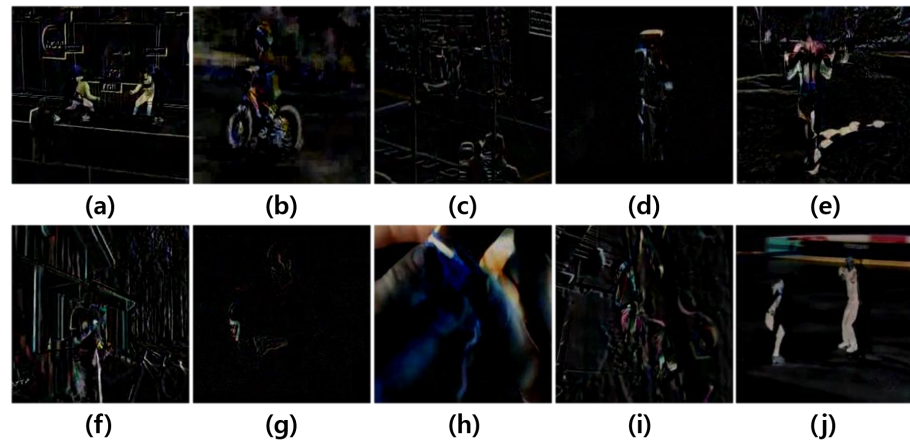


Figure 4. A set of differential images selected from UCF101 dataset: (a) fencing, (b) biking, (c) still rings, (d) pizza tossing, (e) lunge, (f) juggling balls, (g) mixing, (h) knitting, (i) rock climbing, and (j) ice dancing.

The differential image feature map is generated using the differential image as

$$F_{\text{diff}} = \phi \left(I_n^{\text{diff}}(x, y) \right) \quad (4)$$

The differential image feature map returns a bottleneck feature map through the backbone VGG16 network in the same manner as the RGB image feature map.

When learning actions from a video, it is important to train the spatio-temporal information of the action. Therefore, we generated a deep feature map by concatenating RGB image and differential feature maps. Using the extracted feature maps, we computed the feature vector V_n with multiple fully-connected layers as

$$V_n = FC_n(\psi(F_{\text{RGB}} \circ F_{\text{diff}})), \quad (5)$$

where $\psi(\cdot)$ represents a feature map concatenation operator, and FC_n the fully-connected layer for the n -th feature map. Consequently, the generated deep feature map is a combination of two feature maps with spatio-temporal information, and enables spatio-temporal learning in the entire network.

3.3. Bidirectional Exponential Moving Average

In this subsection, we present a bidirectional exponential moving average method to assign a high weight to an important action. In general, a single action in a video consists of a sequence of action elements as shown in Figure 5. More specifically, an action contains preparation, implementation and completion. In this paper, we assume that the most important information is included in the middle, that is, the implementation element.

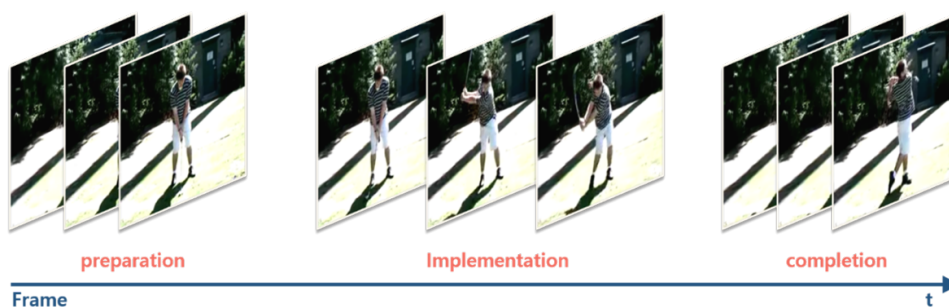


Figure 5. A golf swing action consists of three action elements including preparation, implementation and completion.

Based on that assumption, we assign a higher weight to the middle frames using the bidirectional exponential moving average, which recursively computes the weight as follows

$$S_n = \begin{cases} \alpha S_{n+1} + (1 - \alpha) V_n, & t < \frac{N-1}{2} \\ V_n, & t = \frac{N-1}{2} \\ \alpha S_{n-1} + (1 - \alpha) V_n, & t > \frac{N-1}{2} \end{cases} \quad (6)$$

where S_n represents the value of bidirectional exponential moving average, $\alpha \in [0, 1]$ adjusts the weight of bidirectional exponential moving average. In this work, $\alpha = 0.9$ was experimentally selected.

The final loss value is determined as the mean of the first and last values of S_n as

$$S = \frac{S_0 + S_{N-1}}{2}, \quad (7)$$

where S_0 and S_{N-1} are weighted around the middle feature map sequence in the form of a fully connected layer.

3.4. Late Fusion and Combined Loss

In this subsection, we present a method to improve the accuracy of the proposed algorithm, and prove that the temporal relation matters. The late fusion operation combines previously generated information and the result of C3D, which is the same as the deep feature map using the sampled video clip given in Section 3.1, and the input has a size of $3 \times 16 \times 112 \times 112$.

In this paper, the late fusion operation uses the soft-max value of the bidirectional exponential moving average, based on the deep feature map and the softmax value of C3D.

$$\hat{S} = F_{\text{softmax}}(S) \quad (8)$$

$$\hat{S}_{\text{C3D}} = F_{\text{softmax}}(S_{\text{C3D}}) \quad (9)$$

where S_{C3D} represents the last fully connected layer of the C3D network. To combine the temporal information of the C3D network and the temporal relationship information of the proposed network, each softmax value is fused with the same weight.

$$LF = \frac{1}{2}(\hat{S} + \hat{S}_{\text{C3D}}) \quad (10)$$

The proposed method classifies the action using the bidirectional exponential moving average net using a deep feature map and a late fusion with C3D. Therefore, the loss function performs an operation on each label through least square

$$c = \sum(\text{label} - LF)^2 \quad (11)$$

To reduce the loss value, Adam optimizer was used as the optimization algorithm [14].

4. Experimental Results

4.1. Datasets

In order to learn the spatio-temporal features and temporal relationships, we used action recognition datasets, UCF101 and KTH [13,14]. UCF101 has 101 real actions with challenging conditions such as camera movement, occlusion, and complex background. UCF's 101 action classes consist of 13,320 videos that are divided into five types: human-object interaction, body motion only, human-human interaction, playing musical instruments, and sports. UCF101 uses 9530 images for training and 3790 images for testing. KTH contains six types of action including: walking, jogging, running, boxing, hand waving, and hand clapping. It also includes camera scale and location change to evaluate action recognition performance. KTH has smaller action classes and images than those of UCF101, and consists of four scenarios under six actions.

4.2. Experimental Environment

The proposed algorithm was implemented under the Tensorflow framework. The experimental environment was conducted with Intel Core i7-7700K (4.20 GHz) CPU, 8GB memory, and NVIDIA GeForce GTX 1080Ti. VGG16 was used as the baseline for the experiment, and pre-trained using ImageNet with Adam optimizer for 90,000 iterations and a learning rate of 0.0001.

4.3. Ablation Experiment

In this section, we conducted experiments including an ablation study and an accuracy test of action recognition due to the temporal relation learning proposed in this paper. Table 1 shows the results of ablation studies for UCF101 and KTH datasets. In Table 1, "RGB" represents a backbone using VGG16-net and indicates the result of using the RGB image feature map of the input video clip. "RGB+Diff" represents a deep feature map, which is the result of combining the proposed RGB image feature map and difference image feature map to the backbone in Section 3.3. For each dataset, the performance was improved by 6.2% and 7.4% compared to the previous backbone experiment results. "RGB+Diff+Moving avg." means that a weight is assigned to the action section using the bidirectional exponential moving average neural network proposed in Section 3.4 with the deep feature map.

Table 1. Ablation study of the proposed method.

Model	Accuracy (%)	
	UCF101	KTH Dataset
RGB	48.40	65.28
RGB + Diff	54.62	72.69
RGB+Diff+Moving avg.	55.97	73.10
RGB+Diff+Moving avg. + C3D	72.03	73.61

The performance of "RGB+Diff+Moving avg." was improved by 1.4% and 0.4% compared with the result of the previous step. Lastly, "RGB+Diff+Moving avg.+C3D" is a structure in which all steps of the proposed method in this paper are implemented by the late fusion of C3D. Compared with the previous experiment, this structure improved accuracy by about 16.1% and 0.5%. In conclusion, the experimental results of the proposed method achieved 23.63% and 8.33% higher recognition rates than the backbone performance. Therefore, the results of the ablation study in Table 1 show that the proposed method has an improved accuracy by including spatio-temporal information and temporal relation information.

In order to verify the effect of the bidirectional exponential moving average neural network in action recognition using the proposed deep feature map, an experiment was conducted on parameters that give weights to the main action sections. For the experiment,

the size of the parameter α of the bidirectional moving average neural network was changed based on the proposed deep feature map. As described in Section 3.4, the bidirectional exponential moving average was recursively calculated with $\alpha \in [0, 1]$. The closer to 1.0 α gets, the higher the weight given to the middle frame.

Table 2 shows the increasing accuracy as a high weight of close to 1 is given to the frame, which is the action Implementation section. Therefore, as a result of the experiment, the value of the parameter α was most effective when set to 0.9.

Table 2. Experiment of Bidirectional exponential moving average parameter α .

Moving Average Parameter: α	Accuracy (%)
0.5	53.89
0.6	54.50
0.7	55.19
0.8	55.43
0.9	55.97

4.4. Accuracy

In this section, in order to verify the effectiveness of the proposed method, we compared the accuracy with the other methods, including the baseline. The experiment was conducted in two datasets: UCF101 and KTH dataset. Table 3 shows the action recognition accuracy comparison performed in the UCF101. When comparing the performance in UCF101 with same environment in Section 4.2, 2DCNN, which is a baseline in our method, has a low accuracy of 48.40%. Another model, the Ishan method, also has a low accuracy of 50.90%. In addition, the C3D method which later fused in our proposed method has an accuracy of 70.02%. Therefore, when comparing our proposed method with 2DCNN, Ishan method, C3D, the proposed method showed a 23.63%, 21.13%, 2.01% improved performance.

Table 3. Accuracy comparison in UCF101 dataset.

Model	Accuracy (%)
2DCNN [5]	48.40
Ishan et al. [15]	50.90
C3D [10]	70.02
Proposed method	72.03

Table 4 shows the accuracy comparison performed in the KTH dataset. This experiment shows an increased performance similar to Table 3. When it is performed on the KTH dataset, the recognition accuracy of the 2DCNN is 62.28% and C3D is 66.20%. Therefore, when compared with our proposed method, which has an accuracy of 73.61, the accuracy of our proposed method is improved by 11.33%, 7.41%. Our proposed method is a late fusion of 2DCNN-based Stacked Short-Term Deep features and Bidirectional Moving Average method and C3D, respectively, without deteriorating the existing performance, even though late fusion with C3D was potentially performed.

Table 4. Accuracy comparison in KTH dataset.

Model	Accuracy (%)
2DCNN [5]	62.28
C3D [10]	66.20
Proposed method	73.61

4.5. Comparison of Processing Speed

In this section, in order to verify the processing speed of the proposed method and the corresponding efficiency, an execution speed comparison experiment was performed. This experiment was conducted at the stage before late fusion with C3D in order to prove the effectiveness of the proposed deep feature map and Bidirectional exponential moving average. As shown in Table 5, the processing speed for recognizing an action class in the existing “RGB” is 0.031 seconds, and, in “RGB+Diff” and “RGB+Diff+Moving avg.”, the processing speed is 0.075 seconds. As the proposed method increases, the processing speed of action recognition increases by 0.044sec. However, the general frame rate is 30 frames/s, and it can be seen that the processing speed is processes the action recognition relatively quickly according to the frame rate.

Table 5. Computational complexity comparison.

Model	Accuracy (%)	Speed
RGB	48.40	0.031sec.
RGB+Diff	54.62	0.075sec.
RGB+Diff+Moving avg.	55.97	0.075sec.

5. Conclusions

We proposed a novel action recognition network (ARN) using a short-term pixel-difference and bidirectional moving average. The proposed ARN generates deep features using the short-term pixel-difference image to combine spatio-temporal information and the temporal relationship between adjacent frames. The proposed network gives a higher weight to the middle frames to train important action elements. Finally, the previously generated information and result of C3D are fused to improve the performance of the proposed network. The late fusion result proves that the temporal relationship is important in improving the recognition performance.

Experimental results showed that ARN succeeded in action recognition with a small dataset. A combination of short-term pixel-difference-based deep features and bidirectional moving average significantly improved the performance of the baseline network. Although the ARN additionally takes temporal information into account, it does not require additional computation compared with 2D CNN. As a result, the ARN is suitable for real-time action recognition in terms of both recognition accuracy and computational efficiency.

Author Contributions: J.H. initiated the research and designed the experiments, J.S. and H.P. acquired the test images and performed the experiments, and J.P. wrote and revised the manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported partly by Institute for Information & Communications Technology Promotion (IITP) grant funded by the Korea government (MSIT) [2017-0-00250, Intelligent defense boundary surveillance technology using collaborative reinforced learning of embedded edge camera] and partly by the ICT R & D program of MSIP/IITP [2014-0-00077, Development of global multi-target tracking and event prediction techniques based on real-time large-scale video analysis and supported by the Chung-Ang University Research Scholarship Grants in 2019.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The UCF101 datasets presented in this study are openly available in [13] and can be found here <https://www.crcv.ucf.edu/data/UCF101.php>. The KTH datasets presented in this study are openly available in [14] can be found here <https://www.csc.kth.se/cvap/actions/>.

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

References

1. Dong, J.; Gao, Y.; Lee, H.J.; Zhou, H.; Yao, Y.; Fang, Z.; Huang, B. Action Recognition Based on the Fusion of Graph Convolutional Networks with High Order Features. *Appl. Sci.* **2020**, *10*, 1482. [[CrossRef](#)]
2. Leong, M.C.; Prasad, D.K.; Lee, Y.T.; Lin, F. Semi-CNN Architecture for Effective Spatio-Temporal Learning in Action Recognition. *Appl. Sci.* **2020**, *10*, 557. [[CrossRef](#)]
3. Dong, S.; Hu, D.; Li, R.; Ge, M. Human action recognition based on foreground trajectory and motion difference descriptors. *Appl. Sci.* **2019**, *9*, 2126. [[CrossRef](#)]
4. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. *Commun. ACM* **2017**, *60*, 84–90. [[CrossRef](#)]
5. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
6. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going deeper with convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 1–9.
7. Huang, G.; Liu, Z.; Van Der Maaten, L.; Weinberger, K.Q. Densely connected convolutional networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 4700–4708.
8. Karpathy, A.; Toderici, G.; Shetty, S.; Leung, T.; Sukthankar, R.; Fei-Fei, L. Large-scale video classification with convolutional neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 1725–1732.
9. Ji, S.; Xu, W.; Yang, M.; Yu, K. 3D convolutional neural networks for human action recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2012**, *35*, 221–231. [[CrossRef](#)] [[PubMed](#)]
10. Tran, D.; Bourdev, L.; Fergus, R.; Torresani, L.; Paluri, M. Learning spatiotemporal features with 3d convolutional networks. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 4489–4497.
11. Baker, S.; Matthews, I. Lucas-kanade 20 years on: A unifying framework. *Int. J. Comput. Vis.* **2004**, *56*, 221–255. [[CrossRef](#)]
12. Chaudhry, R.; Ravichandran, A.; Hager, G.; Vidal, R. Histograms of oriented optical flow and binet-cauchy kernels on nonlinear dynamical systems for the recognition of human actions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; pp. 1932–1939.
13. Soomro, K.; Zamir, A.R.; Shah, M. UCF101: A dataset of 101 human actions classes from videos in the wild. *arXiv* **2012**, arXiv:1212.0402.
14. Schuld, C.; Laptev, I.; Caputo, B. Recognizing human actions: A local SVM approach. In Proceedings of the 17th International Conference on Pattern Recognition, Cambridge, UK, 26 August 2004; Volume 3, pp. 32–36.
15. Misra, I.; Zitnick, C.L.; Hebert, M. Shuffle and learn: Unsupervised learning using temporal order verification. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016; Springer: Berlin/Heidelberg, Germany, 2016; pp. 527–544.