algorithms

MDPI

Article

I3D-Shufflenet Based Human Action Recognition

Guocheng Liu [1], Caixia Zhang [2], Qingyang Xu [1,*], Ruoshi Cheng [1], Yong Song [1], Xianfeng Yuan [1] and Jie Sun [1]

[1] School of Mechanical, Electrical & Information Engineering, Shandong University, Weihai 264209, China; 201816503@mail.sdu.edu.cn (G.L.); 201816499@mail.sdu.edu.cn (R.C.); songyong@sdu.edu.cn (Y.S.); yuanxianfeng@sdu.edu.cn (X.Y.); sunj@sdu.edu.cn (J.S.)

[2] Mechanical & Electrical Engineering Department, Weihai Vocational College, Weihai 264210, China; zhangcx1985@163.com

* Correspondence: qingyangxu@sdu.edu.cn

Received: 7 October 2020; Accepted: 13 November 2020; Published: 18 November 2020

Abstract: In view of difficulty in application of optical flow based human action recognition due to large amount of calculation, a human action recognition algorithm I3D-shufflenet model is proposed combining the advantages of I3D neural network and lightweight model shufflenet. The 5×5 convolution kernel of I3D is replaced by a double 3×3 convolution kernels, which reduces the amount of calculations. The shuffle layer is adopted to achieve feature exchange. The recognition and classification of human action is performed based on trained I3D-shufflenet model. The experimental results show that the shuffle layer improves the composition of features in each channel which can promote the utilization of useful information. The Histogram of Oriented Gradients (HOG) spatial-temporal features of the object are extracted for training, which can significantly improve the ability of human action expression and reduce the calculation of feature extraction. The I3D-shufflenet is testified on the UCF101 dataset, and compared with other models. The final result shows that the I3D-shufflenet has higher accuracy than the original I3D with an accuracy of 96.4%.

Keywords: action recognition; 3D convolution; I3D neural network; shufflenet

1. Introduction

With the development of artificial intelligence, the progress of computer vision has received special attention. At present, the world's top scientific research teams and major scientific research institutions are achieving rapid progress in the field of human action recognition. In the 1970s, a human body description model was proposed by Professor Johansson [1], which had a great impact on human body recognition. Video-based human action recognition methods traditionally make use of artificial means to extract motion features. The traditional algorithms for human action recognition consist of Histogram of Oriented Gradients (HOG) [2], Histogram of Optical Flow (HOF) [3], Dense Trajectory (DT) [4] etc. The DT algorithm performs multi-scale division of each frame of the video. After division, the features of each region are obtained by dense sampling based on the grid division method. The time domain feature is extracted to generate the feature of trajectory and then the next trajectory position is predicted when the features of the entire picture are obtained. In recent years, deep learning has developed rapidly. It was widely used in the field of image recognition. With the development of deep learning, it is widely used in the field of human action recognition, which greatly improves the accuracy of human action recognition. Deep learning is a data processing and feature learning method. Through lower-level features extraction of the image, the high-level image features or image attributes can be shaped according to lower-level features, then the human action and movement features can be extracted. At present, the performance of deep learning algorithms for big data is significantly superior to traditional algorithms, and has achieved good performance in computer vision and speech

recognition. Convolutional Neural Network (CNN) was proposed in 2012 because it is appropriate for image processing. However, the temporal features extraction is limited due to the loss of time domain information during the feature extraction for video. The time domain features cannot be effectively extracted by 2D-CNN. Under the background of the great success of convolutional networks in the field of computer vision, graph neural networks (GCNs) based human action recognition become a recent event [5]. Bruna et al. (2013) proposed the important research on graph convolutional networks, and the space-based graph convolutional networks develops rapidly. These methods directly perform convolution on the graph structure through gathering information about neighboring nodes. In recent years, 3D neural networks have been fully developed. The concept of 3D neural networks was first proposed by Gori et al. at 2005 [6] and further clarified by Scarselli et al. at 2009 [7]. The early studies propagating the adjacent information iteratively through the recursive neural network until reaching a stable point, then the representation of the target node is obtained. This process requires massive calculation. Many recent studies are devoted to solving this problem. Ji et al. [8] proposed a three-dimensional convolutional neural network for spatial-temporal features extraction. The 2D-CNN can be extended to 3D-CNN. 3D convolutional neural network (Convolution Neural Network, CNN) has a good effect in comprehensively image features extraction. The original two-dimensional network still extracts the spatial features of static images, and the time domain features is extracted by the third dimension, it can also convolve across multiple frames of images and extract the information before and after the time series. However, with the augment of parameter, the training of the model becomes difficult. A residual network structure was proposed by He et al. [9], which can not only handle the above-mentioned problems caused by increasing the network depth, but also optimize and improve the network performance. Two-stream [10] is dedicated to RGB images and optical flow graphs, then the results of the two are subjected to a fusion. Motion stream (ResNet-50) [11] extracts the spatial-temporal information by introducing the connection between spatial-temporal flow in a two-stream network with fusion blocks. S3D-G (separable 3D CNN) [12] builds an effective video classification system, seeks to strike a balance between speed and accuracy, and replaces many 3D convolutions with low-cost 2D convolutions. MFNet (Multi-Fiber Networks) [13] proposes a Multi-Fiber architecture, which cuts a complex neural network into a lightweight network or a collection of optical fibers running through the network. ARTNet(Appearance-and-Relation Networks) [14] is constructed by stacking multiple general building blocks called SMART to simultaneously model the appearance and relationship of RGB input in a separate and explicit manner. FASTER [15] is used for feature aggregation of spatial-temporal redundancy. Its framework can integrate high-quality representations in expensive models to capture subtle motion information, and the lightweight representations in cheap models to cover scene changes in videos. The existing C3D (3DConvolutional Networks) neural network has a wide application. The I3D (Inflated 3D ConvNet) neural network is improved based on the C3D neural network [16], and the recognition speed and accuracy performance of the network is greatly improved. However, both the C3D and I3D neural network have a traditional way for convolution, some channels may be less useful information and consume the computational power.

The article is inspired by the ideas of GoogleNet and Shufflenet. The original convolution kernel in I3D is replaced by a two-layer convolution kernel, which has the same effect but with lower calculation amount. The channel fusion method is incorporated into the I3D network. The feature extracted by the proposed model will be exchanged with different channel features through shuffle operation [17], and more useful information will be used to improve the performance of human action recognition.

This article mainly includes four parts. The 3D convolutional network is introduced in Section 2, the proposed model is introduced in Section 3, and the Section 4 is the experiments. Finally, it is the conclusion.

2. 3D Convolutional Network

3D Convolutional Neural Network is a popular convolutional neural network applied in the field of human action recognition. 3D neural network can not only convolve two-dimensional images, but also the time sequence. The 3D convolutional neural network has one more dimension than the 2D neural network which can better extract the visual human action characteristics by the three-dimensional convolution kernels.

The convolution process of 2D convolutional neural network can be expressed as:

$$v_{ij}^{xy} = ReLU(b_{ij} + \sum_{m} \sum_{p=0}^{P_i-1} \sum_{q=0}^{Q_i-1} W_{ijm}^{pq} v_{(i-1)m}^{(x+p)(y+q)}) \quad (1)$$

where v_{ij}^{xy} is the i convolution result at the j position in the feature map (x,y) of the layer; $ReLU()$ is the activation function; b_{ij} is the deviation of the feature map; m is the index of the feature map in the layer $i-1$; W_{ijm}^{pq} is the value at the position of the feature map; P_i,Q_i is the width and height of the convolution kernel.

Traditional 2D convolution is suitable for spatial feature extraction, and has difficulty with continuous frame processing of video data. Compared with 2D convolution, 3D convolution adds convolution operation for adjacent time dimension information, which can deal with the action information of continuous video frame, the 3D convolution formula is expressed as follows:

$$v_{ij}^{xyz} = ReLU(b_{ij} + \sum_{m} \sum_{p=0}^{P_i-1} \sum_{q=0}^{Q_i-1} \sum_{t=0}^{T_i-1} W_{ijm}^{pqt} v_{(i-1)m}^{(x+p)(y+q)(z+t)}) \quad (2)$$

where v_{ij}^{xyz} represents the convolution result at the position i of the j feature map of the (x,y,z) layer; $ReLU()$ is the activation function; b_{ij} is the deviation of the W_{ijm}^{pqt} feature map; m is the index of the feature map in the $(i-1)$ layer; is the value at the position (p,q,t) of the feature map, t is the time dimension that is unique to 3D convolution; P_i, Q_i, T_i are the width, height and depth of the convolution kernel.

Traditional deep learning network generally uses a single-size convolution kernel, the input data are processed by the convolution kernel and then a feature set is generated. In the Inception module, convolution kernels with different sizes are adopted to calculate and splice separately. The final feature set no longer has the same uniform distribution, but some correlative features are gathered together and generate multiple densely distributed feature subset. Therefore, for the input data, the corresponding features of the distinguishing region are clustered together after different convolution processing, while the irrelevant information is weakened, resulting in better feature set. The inception structure of I3D in this article is shown in Figure 1 which contains two conv-pool layers and three inception layers.

The I3D network inherits the Inception module of Googlenet, and with different size convolution kernels for feature extraction. According to the idea of Googlenet, one convolution is performed on the previous convolution layer output, and an activation function is added after each convolution layer. By concatenating the two three-dimensional convolutions, more nonlinear features are combined. A branch consists of one or more convolution and pooling. In each Inception module, there are four different branches for the input data. The convolution kernels with different sizes are adopted respectively, and are spliced together finally. The I3D neural network adds a convolution operation for adjacent temporal information, which can complete the action recognition of continuous frame. In order to expedite the deep network training speed, a batch regularization module is added to the network. The network is not sensitive to the initialization, so a larger learning rate can be employed. I3D increases the depth of the network, eight convolutional layers and four pooling layers are used. The size of the convolution kernel of each convolutional layer is $3 \times 3 \times 3$, and the step size is $1 \times 1 \times 1$ respectively, the number of filters is 64, 128, 256, 256, 512, 512. Each convolutional layer is followed by a batch regularization layer, a ReLU layer and a pool layer except for the layers of the conv3a,

conv4a, and conv5a. The kernel size of the first pooling layer is $1 \times 2 \times 2$ and the step size is $1 \times 2 \times 2$. The kernel size and step size of the remaining pooling layers are $2 \times 2 \times 2$, the spatial pooling only works in the first convolutional layer, and spatial-temporal pooling works in the second, the fourth and the sixth convolutional layers. Owe to the pooling layers, the output size of the convolution layers is reduced by 1/4 and 1/2 respectively in space and time domain. Therefore, I3D is suitable for short-term spatial-temporal features learning.

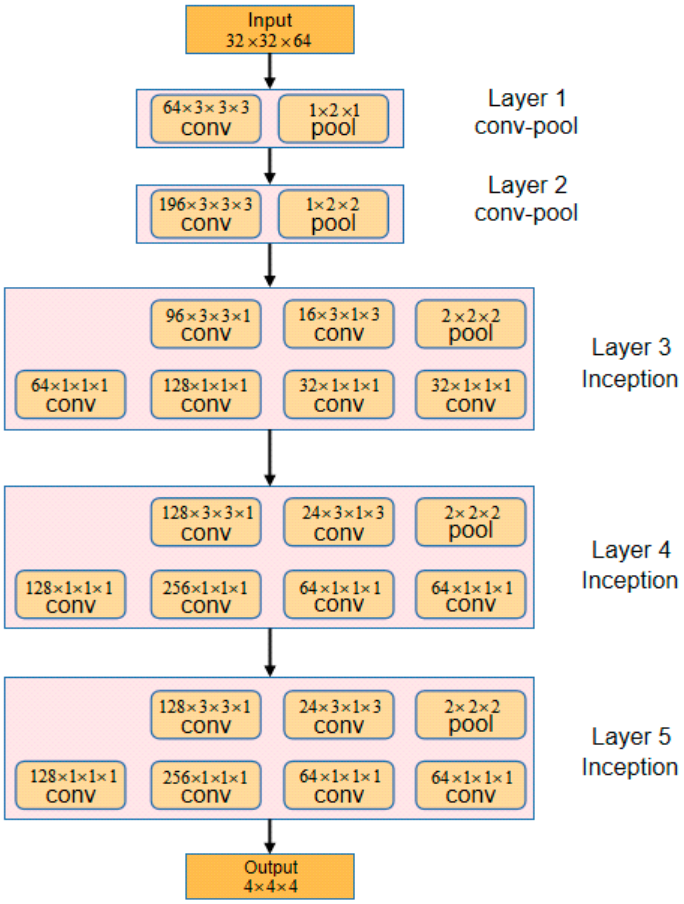


Figure 1. Inception module diagram of I3D.

3. I3D-Shufflenet

3.1. 3D Convolution Kernel Design

Two identical $5 \times 5 \times 5$ convolution kernels are used to perform convolution operations in traditional I3D, which resulting in a large amount of calculation and having the same feature extraction. Through the learning of the convolution kernel, different convolution kernels can be replaced by certain dimensional rules to convolve images through different combinations of convolution kernels. As shown in Figure 2, two-layer $3 \times 3 \times 3$ convolution kernels are used to replace the $5 \times 5 \times 5$ convolution kernel.

Therefore, one of the $5 \times 5 \times 5$ convolution kernel can be replaced by a two-layer $3 \times 3 \times 3$ convolution kernels. The amount of network calculation was reduced by 28% after the change. The structure before and after the replacement are shown in Figure 3. The inception structure in I3D is shown in the Figure 3a, the inception module in I3D-shufflenet is shown as Figure 3b.

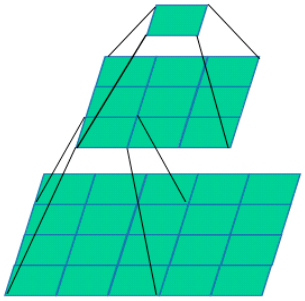


Figure 2. Two-stage 3 × 3 instead of 5 × 5 convolution kernel.

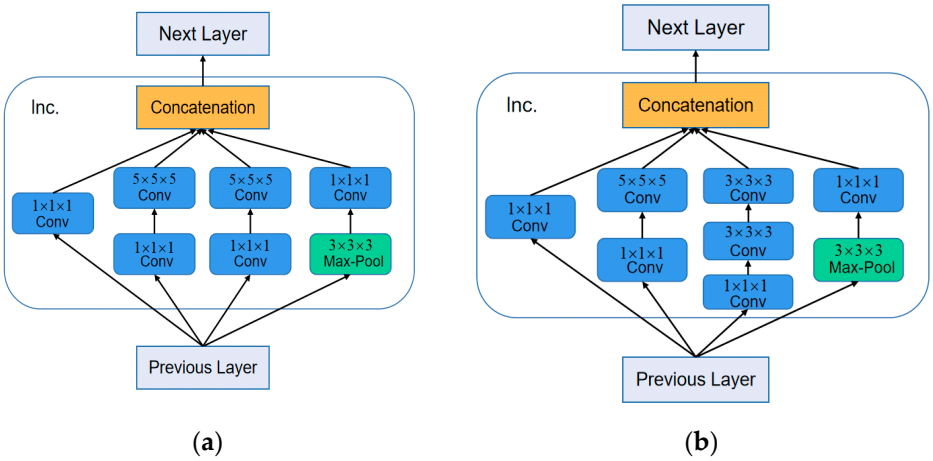


Figure 3. Module before and after the replacement. (**a**) Inception module diagram of I3D; (**b**) Convolution kernel replacement.

3.2. I3D-Shufflenet Network Framework

3.2.1. Channel Shuffle

The channel shuffle draws on the idea of shufflenet. The shufflenet is a deep network proposed by face++. The shufflenet is a computationally efficient CNN model. It is mainly intended to be applied to mobile terminals (such as mobile phones, drones, and robots). Therefore, the shufflenet aims to achieve the best model performance with limited computing resources, which requires a good balance between speed and accuracy. The core operations of shufflenet are pointwise group convolution and channel shuffle, which greatly reduce the amount of calculations while maintaining accuracy. Group convolution is used in I3D neural network. The disadvantage of group convolution is that the output channel is only derived from a small part of input channel. The pixel-level group convolution is introduced for shuffling in order to reduce the computational complexity caused by the convolution operation. Group convolution hinders the information exchange between channels, which will result in a lack of representativeness feature generation. To solve this problem, the channel shuffling is adopted for information fusion between channels.

A channel split operation is introduced. At the beginning of each unit, the c channels of the feature map are divided into two parts: one has c-c' channels and another has c' channels. To minimize the number of shuffles, one part is fixed, and another part contains three convolutions with the same number of channels. Two 1 × 1 convolutions are not grouped in case of the division of two groups second in the channel segmentation. Finally, the features of the two parts are concatenated so that the number of channels maintain fixed and a channel scramble is performed to ensure that the information of the two parts interact.

The residual block is introduced through point-by-point grouping convolution and channel mixing operations. As shown in Figure 4, after point-by-point grouping convolution, the shuffle operation

is performed to improve information flow between different groups on the bottleneck feature of the main branch. Then a smaller $3 \times 3 \times 3$ and $1 \times 1 \times 1$ depth separable convolution is used to reduce the amount of calculation, after a point-by-point grouping convolution, the two branch pixels are added. An average pooling operation is added to the branching for replacing the pixel-by-pixel addition operation which can expand the channel dimension with a small amount of computation.

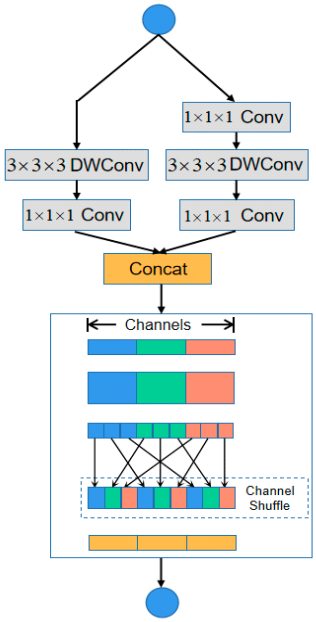


Figure 4. Channel shuffle.

Through the shuffling operation of the 3D convolution layer and combining the Inception-V1 module, the channel fusion network is merged behind the 6th inception of the $3 \times 3 \times 3$ 3D convolution start module in the I3D network. For the I3D model, five consecutive RGB frames are divided into 10 frames and corresponding optical flow segments. The input of the network has 10 frames apart, consecutive RGB frames and corresponding optical flow segments. Through the $3 \times 3 \times 3$ 3D convolutional layer with 512 output channels, the $3 \times 3 \times 3$ 3D maximum merge layer and the complete connection layer, the spatial and kinematic characteristics ($5 \times 7 \times 7$ feature grid, corresponding to time, X and Y dimensions) before the last average merge layer of Inception can be obtained. Therefore, the I3D shuffle network can better extract image features.

3.2.2. I3D-Shufflenet Structure

For the I3D-shufflenet, a two-dimensional convolution is extended to a three-dimensional shufflenet after adding time information. The feature information of the image is fused through different convolution processing of inception, and the output layer of inception is used as the input of channel fusion, in which half of the feature maps are directly entered into the next module. The shuffle operation is added after the 6th layer inception structure in the I3D network, and the shuffle is fused with the image information processed by the 9th layer inception. This can be seen as a feature reuse which is similar to DenseNet and CondenseNet. The other half is divided into three channels by channel segmentation and processed separately. The structure of I3D-shufflenet is shown in Figure 5.

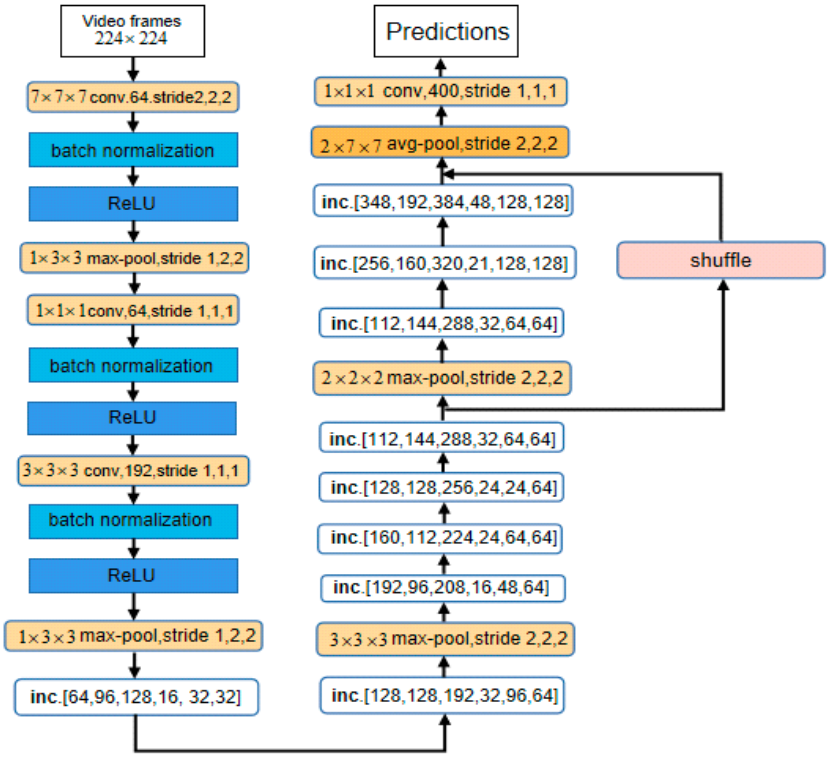
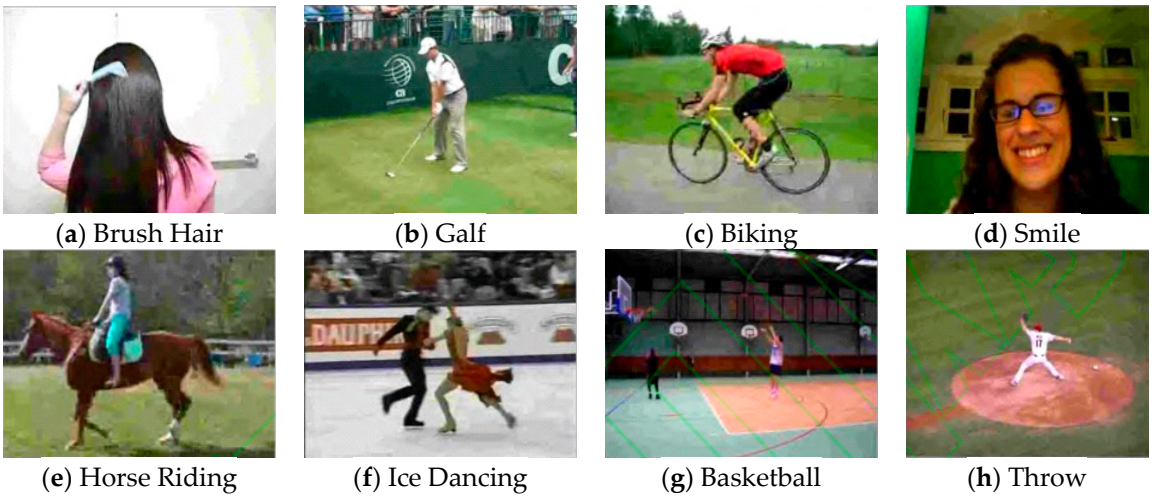


Figure 5. I3D-shufflenet.

4. Experiment

4.1. Data Set for Behavior Recognition

This experiment mainly used the UCF101 data set [18] which is currently the most mainstream data set for human action recognition. The resolution of the UCF101 data set was 320 × 240, and there were 101 types of actions. Each type of action was composed of about six videos taken by 25 people. The 101 human behaviors in the UCF101 data set (total 27 h) were divided into 13320 images with two groups (the training samples and the testing samples) by a ratio of 3:1. Part of the action samples of the UCF101 are shown in Figure 6.



(**a**) Brush Hair (**b**) Galf (**c**) Biking (**d**) Smile

(**e**) Horse Riding (**f**) Ice Dancing (**g**) Basketball (**h**) Throw

Figure 6. Action categories of UCF101.

4.2. Hyperparameter Settings

The rectified linear unit (ReLU) function was used as the activation function in the models. The Adam optimization and the initial learning rate = 0.0001 were used to train the models. The Gaussian weight initialization of convolutional kernels was adopted. In the process of training, the mini-batch was 32. The software and hardware configuration of the experiment were Python3, GPU TitanX and tensorflow on Ubuntu system.

4.3. Channel Shuffle

The Channel Shuffle operation was performed to ensure that the feature branches could exchange information. Group convolution is a very important operation for modern network architectures. It reduces the computational complexity (FLOPs) and changes the dense convolution between all channels (only within groups of channels). On one hand, it allows the usage of more channels under a fixed FLOPs and increases the network capacity with better accuracy. However, the increased number of channels results in more MAC. Formally, the relation between MAC and FLOPs for 1×1 group convolution is:

$$MAC = hw(c_1 + c_2) + \frac{c_1 c_2}{g} = hwc_1 + \frac{Bg}{c_1} + \frac{B}{hw} \tag{3}$$

where g is the number of groups and $B = hwc_1c_2/g$ is the FLOPs. Therefore, given the fixed input shape $c_1 \times h \times w$ and the computational cost B, MAC increases with the growth of g.

At the beginning of unit, the input of c feature channels were split into two branches with c-c' and c channels respectively. First, the high efficiency in each building block enabled using more feature channels and larger network capacity. Second, in each block, half of the channel feature (when $c' = c/2$ went through the block and joined the next block directly. This can be regarded as a kind of feature reuse. The number of "directly-connected" channels between i-th and $(i + j)$-th building block was $r^j c$, where $r = (1 - c')/c$. In other words, the amount of feature reuse decayed exponentially with the distance between two blocks, and the feature reuse became much weaker between distant blocks. Therefore, it reached the lower bound of the value under the given FLOPs when the number of input characteristic channels and the number of output characteristic channels were equal $c_1 = c_2$. Therefore, the three-channel number was adjusted in the program as shown in Figure 7 to better process image features with higher efficiency.

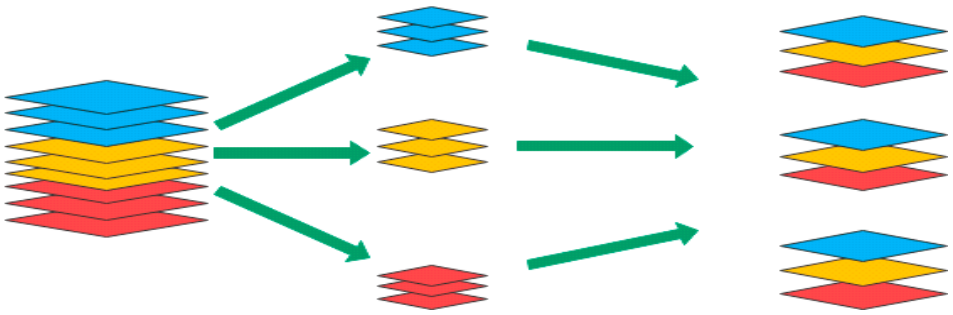


Figure 7. Channel Fusion.

4.4. Loss Function

The loss function is very important to measure the model training, which can evaluate the performance of the model and make corresponding changes to the model. In this paper, the cross-Entropy loss function is adopted as the loss function, the formula is as follows:

$$H(p,q) = -\sum p(x) \log q(x) \tag{4}$$

Among them, p is the correct answer, and q is the predicted value. The smaller the cross entropy is, the closer the probability distribution is. On this basis, the Softmax function is adopted to calculate the probability of each class, the formula is as follows:

$$S_x = \frac{e^{c_x}}{\sum_y e^{c_y}}, \forall x \in \{1, 2, \ldots, N\} \tag{5}$$

Among them, S is the score of the classification probability of each result N. Suppose a problem has a classification problem with N possible results. When an input image is classified, the classification score c is obtained according to each result.

After the prediction of the model, the actual class with the minimum loss value and the probability of this class was the largest. The loss diagram is shown in Figure 8, which shows the improvement of adding channel fusion, the I3D-shufflenet converged faster.

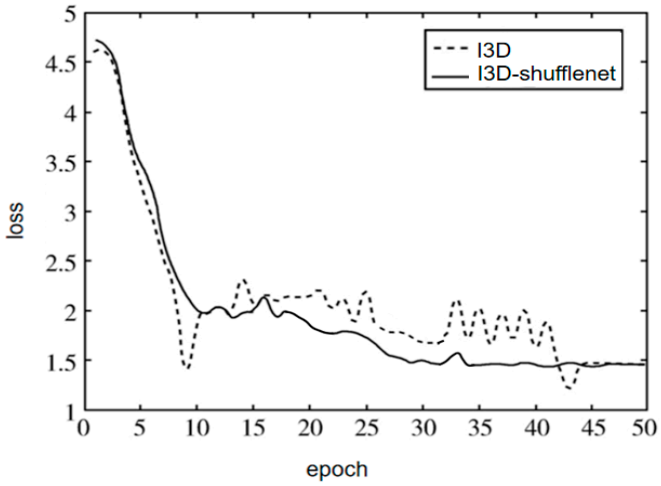


Figure 8. Diagram of loss.

4.5. Learning Rate Setting

For the Adam optimization, two different initial learning rates (0.001 and 0.0001) were used to train the model, respectively. Through experiment, we found that the model with the initial learning rate = 0.0001 had better convergent performance. The model with the initial learning rate = 0.0001 obtained an accuracy of 95%. An exponential decay was used to adjust the value of the learning rate. In other words, the learning rate decreased continuously according to the number of the iterations.

$$\alpha = 0.95^{epoch\_num} \bullet \alpha_0 \tag{6}$$

where *epoch_num* is the current number of iterations; α_0 is the initial learning rate.

The accuracy of the first 50 iterations of the original I3D and the I3D-shufflenet are shown in Figure 9. It can be seen that the I3D-shufflenet had higher accuracy than the I3D after the 15th iteration.

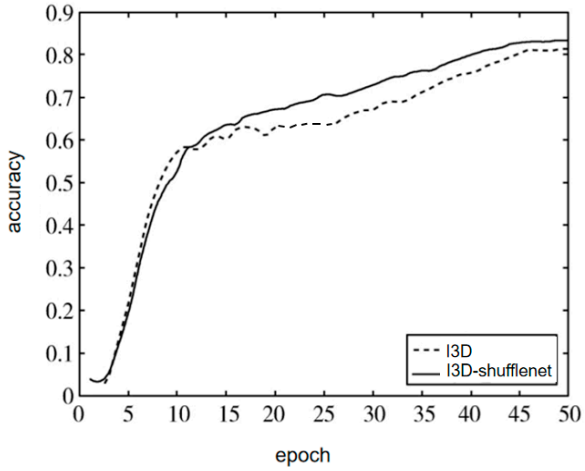


Figure 9. Diagram of accuracy.

Figure 10 presents the confusion matrix for I3D and the I3D-shufflenet.

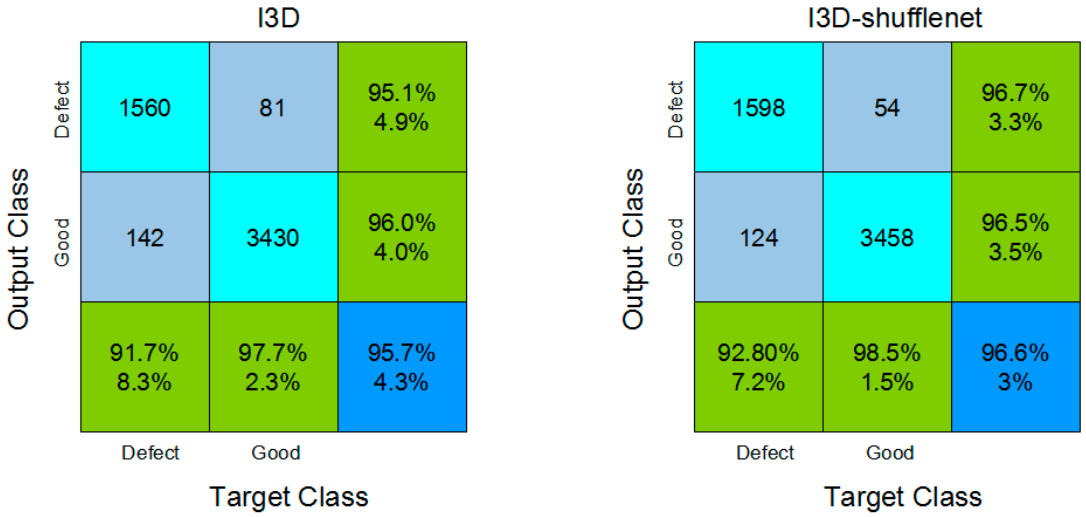


Figure 10. Confusion matrix of I3D and I3D-shufflenet.

Table 1 presents the recall, precision, Area under the ROC (Receiver Operating Characteristic) Curve (AUC) and F1 score for I3D and the I3D-shufflenet. Figure 11 presents the ROC curve [19–21] of I3D and the I3D-shufflenet.

Table 1. Training time comparison of I3D-shufflenet and other networks (h).

Approach	Overall Accuracy	Precision (Defect Class)	Recall (Defect Class)	F1 Score (Defect Class)	AUC (Defect Class)
I3D	95.7	0.9506	0.9166	0.9336	0.9463
I3D-shufflenet	96.6	0.9673	0.9280	0.9477	0.9641

4.6. Feature Map Output

The boxing and Taiji examples are selected for the feature extraction exhibition. The feature map extracted by normal I3D and I3D-shufflenet is shown in Figure 12. According to Figure 12, I3D had some limitations for continuous action feature extraction, a lot of key action information was lost. I3D-shufflenet made use of the shuffle operation, more action information was captured, and the characteristics of feature map were more obvious.

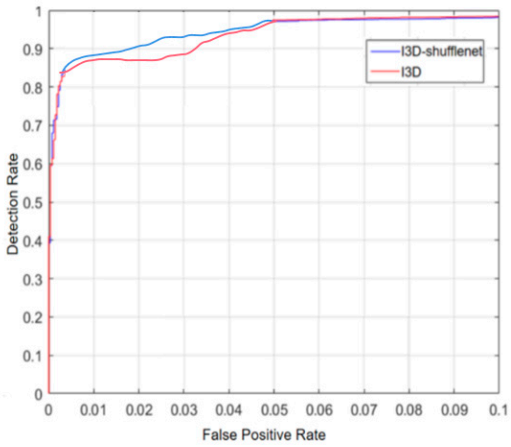


Figure 11. ROC curve for I3D and I3D-shufflenet.

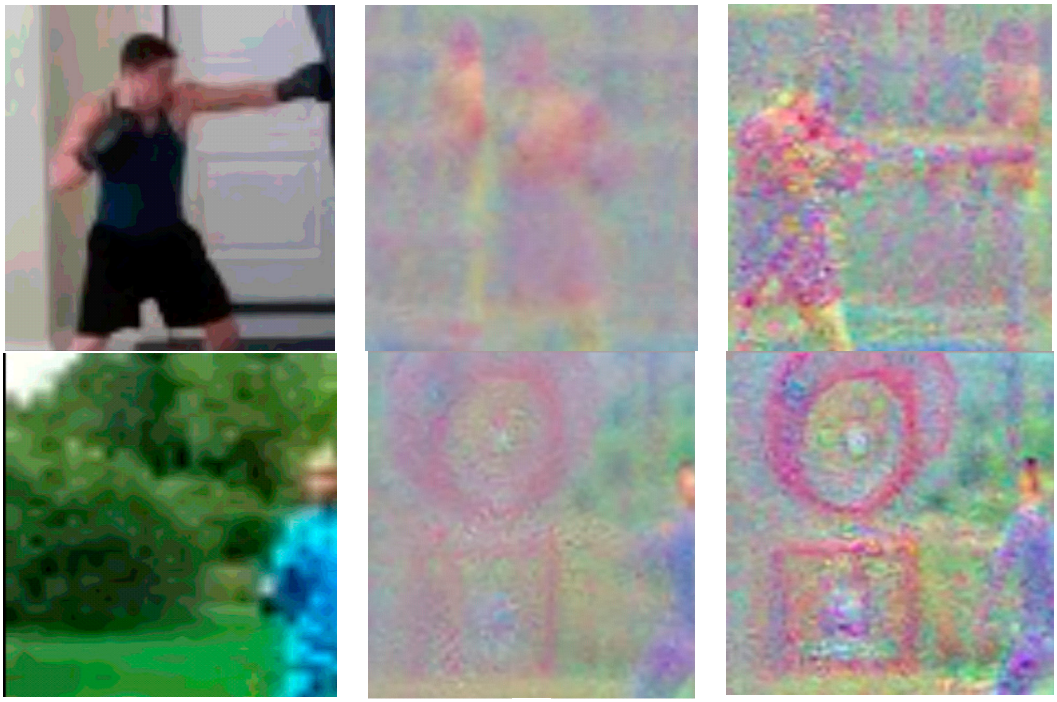


Figure 12. Feature map of models.

4.7. Class Activation Mapping

Figure 13 shows the CAM (Gradient-weighted Class Activation Mapping) [20] result obtained from boxing and Tai Chi videos. The figures show the important features for action recognition. The distinguishing area was the action part, which helped the I3D network to determine. Different cases can be obtained in [13].

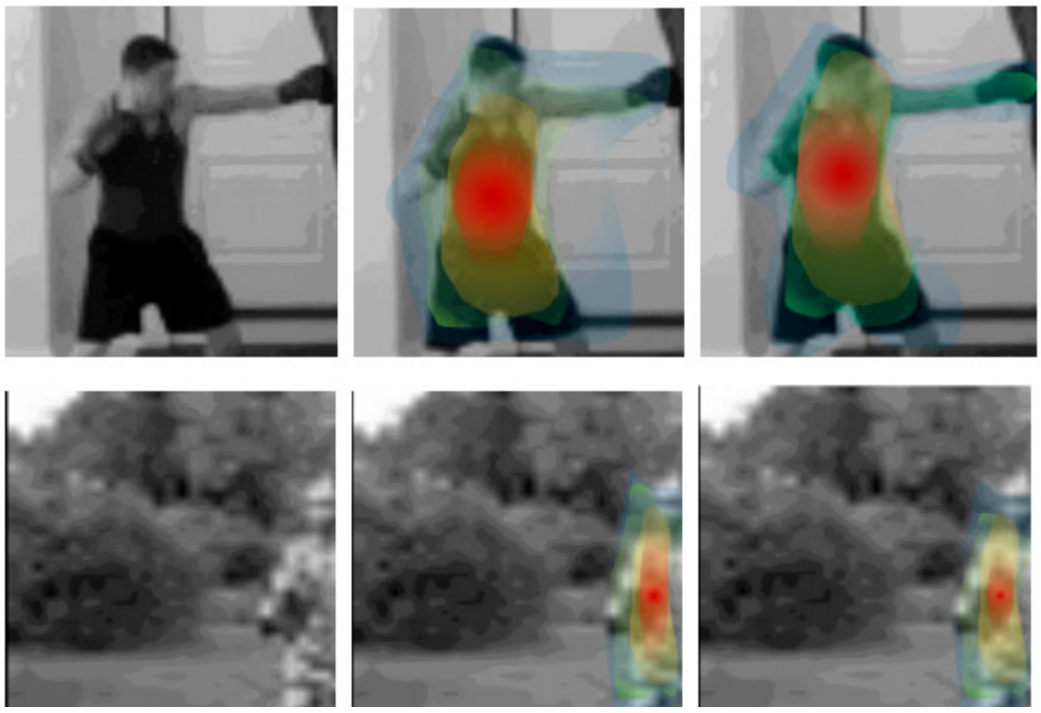


Figure 13. Typical CAM output of boxing and Tai Chi.

4.8. Comparisons

Compared with the I3D, the training time of I3D-shufflenet on the UCF101 dataset was reduced by 15.3% under the same settings. The running time of the of typical human action recognition models are shown in Table 2. The current accuracy of the neural networks on the UCF101 dataset are shown in Table 3.

Table 2. Training time comparison of I3D-shufflenet and other networks (h).

Model	UCF101
C3D	16.5
P3D	29.2
R3D	30.7
I3D	26.1
I3D-shufflenet	22.3

Table 3. Human action recognition comparison of different algorithms on UCF101 (%).

Algorithm	Accuracy
Two-stream [10]	88.0
Motion stream(ResNet–50) [11]	87.0
S3D-G [12]	96.8
MFNet [13]	96.0
ARTNet [14]	93.5
FASTER32 [15]	96.9
Two-stream(conv fusion) [22]	92.5

Table 3. *Cont.*

Algorithm	Accuracy
Two-stream(SI+OF) [23]	93.9
C3D [16]	82.3
IDT [24]	85.9
TSN [25]	94.9
R(2+1)D BERT [26]	98.7
I3D	95.6
I3D-shufflenet	96.4

5. Conclusions

This article mainly studies the improvement and enhancement of channel fusion for the original I3D neural network. The channel shuffle module is added to the inception module of the I3D network. The original channel number is divided into three channels, and the shuffle operation is used to improve the I3D network's recognition accuracy by splitting and reorganizing the channels to better extract image information. In addition, this paper improves the convolution kernel of the Inception module in the I3D neural network. The training speed of the I3D is improved without performance declining, and the performance of the proposed model is better than I3D. Further improving the model and making use of information fusion will be our future work.

Author Contributions: Methodology, G.L.; formal analysis, C.Z., X.Y. and J.S.; supervision, Q.X. and R.C.; project administration, Y.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Science and Technology Major Project (NO. 2018ZX01031201, 2018ZX09201011).

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

References

1. Johansson, G. Visual motion perception. *Sci. Am.* **1975**, *232*, 76–89. [CrossRef] [PubMed]
2. Žemgulys, J.; Raudonis, V.; Maskeliūnas, R.; Damaševičius, R. Recognition of basketball referee signals from videos using Histogram of Oriented Gradients (HOG) and Support Vector Machine (SVM). *Procedia Comput. Sci.* **2018**, *130*, 953–960. [CrossRef]
3. Li, T.; Chang, H.; Wang, M.; Ni, B.; Hong, R.; Yan, S. Crowded Scene Analysis: A Survey. *IEEE Trans. Circ. Syst. Vid.* **2015**, *25*, 367–386. [CrossRef]
4. Wang, H.; Klaser, A.; Schmid, C.; Liu, C. Action Recognition by Dense Trajectories. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, 20–25 June 2011; pp. 3169–3176.
5. Kipf, T.N.; Welling, M. Semi-Supervised Classification with Graph Convolutional Networks. *arXiv* **2016**, arXiv:1609.02907.
6. Gori, M.; Monfardini, G.; Scarselli, F. A New Model for Learning in Graph Domains. In Proceedings of the 2005 IEEE International Joint Conference on Neural Networks, Montreal, QC, Canada, 31 July–4 August 2005; pp. 729–734.
7. Scarselli, F.; Gori, M.; Tsoi, A.C.; Hagenbuchner, M.; Monfardini, G. The Graph Neural Network Model. *IEEE Trans. Neural Netw.* **2009**, *20*, 61–80. [CrossRef] [PubMed]
8. Ji, S.; Xu, W.; Yang, M.; Yu, K. 3D Convolutional Neural Networks for Human Action Recognition. *IEEE Trans. Pattern Anal.* **2013**, *35*, 221–231. [CrossRef] [PubMed]
9. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.

10. Simonyan, K.; Zisserman, A. Two-Stream Convolutional Networks for Action Recognition in Videos. In Proceedings of the 27th International Conference on Neural Information Processing Systems, Montreal, QC, Canada, 8–13 December 2014; pp. 568–576.
11. Feichtenhofer, C.; Pinz, A.; Wildes, R.P.B.I. Spatiotemporal Multiplier Networks for Video Action Recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 7445–7454.
12. Xie, S.; Sun, C.; Huang, J.; Tu, Z.; Murphy, K. Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In Proceedings of the 15th European Conference. Proceedings: Lecture Notes in Computer Science (LNCS 11219), Tokyo, Japan, 29 October–2 November 2018; pp. 318–335.
13. Chen, Y.; Kalantidis, Y.; Li, J.; Yan, S.; Feng, J. *Multi-Fiber Networks for Video Recognition*; Lecture Notes in Computer Science; Springer: Cham, Switzerland, 2018; pp. 364–380.
14. Bilen, H.; Fernando, B.; Gavves, E.; Vedaldi, A. Action Recognition with Dynamic Image Networks. *IEEE T Pattern Anal.* **2018**, *40*, 2799–2813. [CrossRef] [PubMed]
15. Zhu, L.; Tran, D.; Sevilla-Lara, L.; Yang, Y.; Feiszli, M.; Wang, H. FASTER Recurrent Networks for Efficient Video Classification. In Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence (AAAI-20), New York, NY, USA, 7–12 February 2020; Volume 34, pp. 13098–13105.
16. 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.
17. Zhang, X.; Zhou, X.; Lin, M.; Sun, J. ShuffleNet: An Extremely Efficient Convolutional Neural Network for Mobile Devices. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 6848–6856.
18. 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.
19. Narayanan, B.N.; Beigh, K.; Loughnane, G.; Powar, N. Support Vector Machine and Convolutional Neural Network Based Approaches for Defect Detection in Fused Filament Fabrication. *Int. Soc. Opt. Photonic* **2019**, *11139*, 1113913.
20. Narayanan, B.N.; Ali, R.; Hardie, R.C. Performance Analysis of Machine Learning and Deep Learning Architectures for Malaria Detection on Cell Images. *Int. Soc. Opt. Photonic* **2019**, *11139*, 111390W.
21. Narayanan, B.N.; De Silva, M.S.; Hardie, R.C.; Kueterman, N.K.; Ali, R. Understanding Deep Neural Network Predictions for Medical Imaging Applications. *arXiv* **2019**, arXiv:1912.09621.
22. Feichtenhofer, C.; Pinz, A.; Zisserman, A. Convolutional Two-Stream Network Fusion for Video Action Recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 1933–1941.
23. Wang, L.; Li, W.; Van Gool, L. Appearance-and-Relation Networks for Video Classification. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 1430–1439.
24. Bonanomi, C.; Balletti, S.; Lecca, M.; Anisetti, M.; Rizzi, A.; Damiani, E. I3D: A new dataset for testing denoising and demosaicing algorithms. *Multimed. Tools Appl.* **2018**, *79*, 8599–8626. [CrossRef]
25. Wang, L.; Xiong, Y.; Wang, Z.; Qiao, Y.; Lin, D.; Tang, X.; van Gool, L. Temporal segment networks: Towards good practices for deep action recognition. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 8–16 October 2016; pp. 20–36.
26. Kalfaoglu, M.E.; Alkan, S.; Alatan, A.A. Late Temporal Modeling in 3D CNN Architectures with BERT for Action Recognition. *arXiv* **2020**, arXiv:2008.01232v3.