Multi-Person Action Recognition Based on Deep Learning

Sen Wang
School of ECE
Georgia Institute of Technology
Atlanta, GA
swang736@gatech.edu

Ziyi Zhou
School of ECE
Georgia Institute of Technology
Atlanta, GA
zzhou387@gatech.edu

***Abstract*—We proposed one action recognition algorithm aimed at person-level detection with fast speed. The algorithm is composed of three parts: person-detection, action sequence connection, and individual person action recognition. The main contributions are fourfold. First, different from popular methods, we use the picture detection results of the first step for the following analysis. We made use of the continuity of coordinates of bounding boxes, and use interpolation to improve the speed of person detection between frames a lot with little loss of accuracy. Whats more, we also improved the original YOLO algorithm on human detection in low-resolution situations. Second, we proposed new similarity criteria to connect the person-level sequence to achieve better effect. Third, we designed a fast action recognition network after comparing with different nets, and argued that our network could achieve the best balance between training cost, testing cost and accuracy.**

***Index Terms*—spatial action recognition, deep learning, cost-efficient sampling, human detection**

I. Introduction

The faster and faster development of computer science and artificial intelligence proposes higher requirement for image processing and video analysis, among which action recognition, especially those aiming at spatial and temporal action analysis, occupies an important position. The algorithm with higher accuracy and real-time processing speed could create enormous commercial appliances, and bring amazing changes to our daily life. One of the main challenges of action recognition is about algorithm computational complexity, or calculation speed. State-of-the-art work could achieve good performance, which means, accuracy more than 90% and processing speed near or higher than 40 FPS, however, almost all of these algorithms are tested on those high-standard, which means, expensive, hardware equipment, like titan GPU, and so limits the actual appliances of these algorithms into real world. In this paper, we present a new multi-person action recognition algorithm based on existing research works, which could achieve high accuracy with faster calculation speed, and which we believe could bring some beneficial inspiration to the industrial. Several new techniques are introduced in the paper to improve both accuracy and speed. The first one is about picture-transfer mechanism. Generally speaking, to achieve multi-person action detection, we need to detect all the person shown in the video, and then analyze their action individually. Both of these two steps can be achieved by well-designed deep learning network, which, however, requires information transfer. Previously, when concerning such problems, feature map, rather than original pictures, is usually a good choice, because people think both of these two networks just do the similar work and so conveying feature could save computation. However, since the following network which analyzes peoples action sequence requires the input matrix to be the similar size, the feature map generated by the first network needs to be resized by ROI pooling [1] or by other methods, which is not by equal proportion, and will inevitably cause the loss of useful information. Therefore, we propose to transfer the picture region of interest (ROI) instead of the feature map of ROI, which we believe could bring some improvements to the final detection accuracy. The loss of computation speed could be compensated by making use of the characteristics of picture ROI that their spatial coordinates are continuous between two frames. As such we dont need to analyze all the frames in one video sequence, but only some percent of them, and the frames between the sampled ones could be calculated by interpolation. We could improve calculation speed a lot in this way, and we found that a sampling rate of about 30% is enough to provide good positioning accuracy. Furthermore, inspired by subsampling in skip-gram model [34], we proposed to use a weighted sampling method to improve the speed and sampling efficiency even further by setting a higher sampling rate on those clips with fast changing speed and lower sampling rate on those clips with slow changing speed.

Second, one of the popular action detection algorithms is the two-stream network which is proposed in 2014 [2]. It analyzes the spatial information and temporal information from two network respectively, and then fuse the class scores together. The temporal network generally takes dense optical flow images as input, though it requires big computation slows down the total calculation process. As a result, it is better to find an alternative way to analyze the temporal information. Based on the current work, we tried to use RNN and one decomposition of C3D respectively to extract the sequences dynamic pattern. With principles of low calculation cost, we designed one small temporal net at the top of the CNN net, and merge the temporal and spatial analyzation to get the final results.

The other techniques include the improvement on low resolution human detection of YOLO by improving the NMS step, include the difference of feature map for better connection accuracy in sequence connection, and using the expanded bounding boxes to include more background information to help achieve higher action recognition results.

To summarize, we are trying to detect all the persons action in a given video. The total algorithm is mainly composed of three steps: first, we need to detect all the people in the video in frame level; then we will connect the action sequence for each person, producing a set of sequence which has only one person and few background pixels. The length of each sequence may be different, depending on the situation. Finally, all the sequences will be analyzed one by one, which after our experiments we suggest to use one simple ResNet too detect their action.

The main contribution of this paper is summarized below: First, we propose to use part of the original pictures rather than parts of feature map as the representative of the people in the video. Such a way is beneficial to convey more precise useful information to the next step, and so improve the total performance of the algorithm. The additional calculation cost could be compensated by our proposed sampling-interpolation mechanism which improves human-detection in videos a lot. Second, we designed some low-calculation temporal network, which could achieve not only good recognition result, but also high speed.

II. Related Work

Recently, since the amazing performance that deep learning methods achieve in computer vision domain, most existing work is built on well-designed CNN network. According to the main components of our algorithm, we make a review of related work.

A. Object Detection

As the development of ImageNet Challenge, proposed by Feifei Li [3], object detection and image processing domain attracts a large amount of researchers participation, bringing much amazing work on the design of the network [4]–[9]. Girshick R and his team proposed a series of algorithms, including RCNN, Fast RCNN, Faster RCNN, for object detection [8], [10], [11], which have a big influence in this area. It achieves object detection by two steps: objects positioning and object classification. A fully convolutional Region Proposal Network (RPN) is used to generate many anchor boxes, which will then be scored for classification and regressed for better positioning accuracy. J Redmon and his team proposed YOLOwhich uses direct regression to detect the objects in the picture [9], and so to be operated at a faster speed. Then Redmon adopts many inspirations to improve the performance of YOLO [12], [13], like adding the Anchor mechanism, coordinate transformation of bounding box, prediction across different scales, and finally achieves state of the art performance in both speed and accuracy. In our work, we mainly use it to detect person, and so make some necessary changes to suit our problem situation and improve the calculation speed for videos.

B. Action Recognition

There is so much excellent work based on deep learning methods for action recognition in these years, and we think most of them could be divided into three categories: two-stream network and its variants, extensions on the temporal dimension of object detection algorithms, three-dimension convolutional network and its changes or combinations with other methods. Many algorithms about action recognition are based on the idea of two-stream network [2]. The information contained in one action sequence could be divided into two parts: spatial information and temporal information. The spatial information could be analyzed from each frame individually, and the temporal information requires the combination of several adjacent frames to extract the changes of moving people. Many algorithms analyze the temporal information from stacked optical flow images [14]–[16] which require high calculation cost. Whats more, based on some experiments, some argue that optical flow of the two-stream network relies on the appearance invariance rather than temporal information to be useful for action recognition [17]. In other words, the temporal information which is supposed to be made full use of is still waiting for development in two-stream methods. There are also many algorithms trying to combine the idea of two-stream network with other methods, like C3D [18]–[21], Recurrent neural network (RNN) [22], RPN [23], [29], and so on. RNN could be used directly for action recognition problem [30], whose structure is suitable for extracting temporal information in videos. Most of them show good performance on scene action classification problems. In our work, we made many tests to combine the idea of two-stream network with ResNet and the above-mentioned temporal network, and showed their results in Experiments section. Since the big similarity of the two problems, object detection and action recognition, many consider to extend the top performing object detection algorithms into three dimensions to achieve action recognition. One direction is to deal with frame tubes rather than ROI of images individually, and some related work, like ACT model [24], T-CNN model [25], two-stream based T-CNN method [26], shows good performance in detection accuracy. There are many different ways to generate action tubes, but most of them face some difficulty of different degree to find the action border if one action sequence contains different action, which could never be ignored in actual world. Some also consider to generate action proposals from low-level cues, like dense trajectories [31]–[33]. However, features obtained from such ways usually cannot be adjusted in a flexible way, and so there is less space for improvement. As a result, we still decide to deal with the input video frame by frame to get higher temporal localization accuracy. Another one important research direction is three-dimension convolutional network (C3D) [27], [28]. It adds the temporal dimension to the convolutional kernel, and so could be applied to videos directly. C3D method could calculate at faster speed, and so

is favored by some companies. However, C3D also faces one big problem that it requires especially large computation cost to train its network, which means not only a larger training set but much more memory and epochs. Both of these two requirements are not easy to be satisfied in actual world, and so many seek to combine the idea of C3D with other methods [19]–[21], like reducing the three-dimension convolution into one two-dimension convolution and one temporal convolution. Such method could reduce the training difficulty of C3D, as well as the calculation cost, so we also give it a try in the Experiments section.

III. Methodology

We propose one fast multi-person action recognition algorithm there, which takes one video sequence as input, and output all the persons position in the image and their action in the video. The process has three steps: detecting each person in each frame, generating the action sequence of each person, recognizing the action of each sequence. The following sections will introduce them in detail.

A. Human Detection

We mainly used YOLOv3 to detect the people shown in the video. There is no problem if we apply YOLO directly to find all the person frame by frame, however, wasting unnecessary computation there is not a wise choice. Different from generating feature matrices for each detected person in each frame, we just used the detected bounding box coordinates to represent each person, from which we gained more precise information.

Sampling-Interpolation Generally speaking, since the especially small time interval between two frames in one video, the recorded peoples action could be seen as continuous. The change of action itself and peoples bounding box coordinates are also very small thanks for the high FPS in real videos, and could also be seen as continuous in most situations. As a result, there is no need to analyze each frame to find the bounding box locations of all the person, we could set one sampling rate, and only calculate the location of these selected frames. The rest could be gotten by interpolation. Interpolation requires much less calculation cost comparing with normal analyzation process, as such we could reduce the average calculation cost of each frame a lot by this way. However, if we hope to get the feature matrixes of each bounding boxes rather than bounding boxes coordinates, such a way could not be applied since we do not know the changing pattern of them. The total process is based on the assumption that the output is almost continuous between adjacent frames, which we cannot guarantee for feature matrixes. An improved sampling mechanism is inspired by subsampling of skip-gram model in NLP, we can use a weighted sampling rate to achieve higher accuracy and computation cost even further. We know adjacent frames with little changing rate could be considered to sample with bigger sampling rate since they do not change a lot, and vice versa. The changing rate of adjacent frames could be evaluated in the way introduced in the second part:

$$SamplingRate \propto \frac{1}{\frac{d}{dt}Similarity} \tag{1}$$

Improved NMS We also made one small improvement from the initial YOLOv3. YOLO3 used the objectness score to choose the best bounding box to represent the object. Such a way is simple and effective for multi-class object detection, however, there is some space to improve if we only want to detect one class, human, in our situation. After the analyzation for some error examples, we found there are many bounding boxes with high classification score but low objectness score, and such bboxes are usually very blurry caused by frames. The blurry effect caused by videos is beneficial for visual effect, but it makes the problem different from original object detection in pictures. As such, we proposed to use a weighted summation of both classification score and objectness score to detect human. The idea is verified partly in our experiments. Some wrong detection results may contain some very wrong results, but such detection results could be removed in the following sequence connection step.

Expanded equal size bboxes We used expanded bounding boxes to get equal size bboxes result for the convenience of action recognition network input. Furthermore, it include more background information and so could provide more helpful information for the following action recognition network. Since the following network for action recognition needs the input to be equal size, we need to transform the output matrixes to be of the same size. One easy way to achieve this is to directly stretch the detected bounding boxes to the same size, which, however, would change the initial resolution. If all the pictures just change from initial resolution in the same way, then it will not add difficulty for the following recognition network since the feature of the same class is consistent in both the training process itself and testing process. Therefore, although people in the frames are of different sizes, we can still make them suitable for the following network following two steps. First, making small changes of the length and width of bounding boxes to stretch them into the same ratio of length to width, like 2.5:1. Secondly, since all the people are of the same size, we could directly stretch them from a ratio of 2.5:1 to 1:1, which is required by the action recognition net. If some bounding boxes edges cannot be changed, for example, someone appears in one corner of the picture, then we will move the center of the bounding boxes to the extent that we get the output bounding boxes satisfying our requirements with the minimum changes from the initial YOLO detection. The feature transmission process also faces some difficulty there. To get the same output size, one usual way is to use ROI to cut the feature matrices. Although such a way could achieve good results in some situation, it still throws away much useful information in a way that cannot be explained perfectly, and so we think avoiding it when possible is beneficial for the total algorithm.

B. Action Sequence Generation

After analyzing the video in 3.1, we could get all the persons bounding boxes in each frame. The next step is to try to generate each persons action sequence. The input of this process is one or several bounding boxes of each frame in the video, and the output is a set of action sequence, each one of which only contains one person and little background information.

Similarity between two bounding boxes To generate the action sequence, we first need to evaluate which two bounding boxes belong to one person. So we define the similarity between two bboxes as below:

$$Similarity = \lambda_{coor} \times IOU(i,j) + \lambda_{obj} \times \sqrt{\sum_{m=0}^{79} (lg(class_{im}) - lg(class_{jm}))^2} \tag{2}$$

The first item is Intersection over Union (IOU) of the two bounding boxes i, j. When finding the match for one sequence, only the last frame of the sequence is used to calculate IOU. The Euclidean distance of the class scores of two bounding boxes represents the second item. When concerning with sequence, all the frames of the sequence are considered since this item mainly represents spatial similarity. The final similarity of these two bounding boxes is the weighted summation of these two items. Based on the continuous assumption mentioned before, the first item assumes the bounding boxes of two frames that locates in the nearest place belong to one persons action sequence, no matter this person is moving or not. In some rare situations, like two people are overlapping, such criteria cannot guarantee to find the correct match, however, it is true for most situations and so we think it is still helpful for our problem. The second item is not used widely in other work. Since YOLO could generate the 80 class scores for each bounding box, we could make use of it by seeing these 80 object classes as 80 dimensions of the objects appearance. Then, the 80 class scores could constitute one feature vector to represent the bounding boxs appearance information. Considering that YOLO usually generates class scores distributing in a wide range, like 10^{-6}-0.99, we used the logarithmic result of the class score as the feature vector. The coefficient of the second item decides the importance of appearance similarity in the connecting process. The higher coefficient means the existing sequence and the next item must have high similarity, and so the changing of appearance, especially those happen when the people change the action, could be caught. However, using such a criteria to judge the transition of action is not comprehensive since it does not consider the temporal information, and so based on the similarity criterion, we define the speed of action as the speed of the change of the second item. This item could be used as the third item of the similarity function, and we also give it one coefficient to control its influence of the total similarity, which is just decided from experiments.

$$Similarity_{new} = \lambda_{coor} \times IOU(i,j) + \lambda_{obj} \times \sqrt{\sum_{m=0}^{79} (lg(class_{im}) - lg(class_{jm}))^2} + \lambda_{speed} \times \frac{d}{dt} Similarity \tag{3}$$

Based on the similarity defined before, we can connect the bounding boxes of different people in frames into a series of action sequence that only contains one person respectively. The detected person in the first frame initiate each sequence individually, and then these sequences will find their match based on the highest similarity which is defined before to find its match. There is one threshold in case the sequence needs to be ended because all the bounding boxes in the next frame are not good matches. However, when all the sequences in the previous frame are matched or ended and there are still bounding boxes in the next frame, then these bounding boxes will begin a new sequence. After all the frames are analyzed, the sequence with short length will be decided to abandon or reconnect, following the same algorithm but with a slightly lower threshold.

C. One-person action recognition

The task of the final step is to analyze the action of each sequence, which has only one person with one action and little background information. Before the detailed introduction of the algorithm, I think we should consider one problem first: is there any meaning to use such a way to recognize action. The cut part of image only contains a person with little background. The little background information adds a lot of difficulty for human to recognize since we mainly judge peoples action or not from the relative moving to the environment. But we think it is not the same for computers. When watching videos, people could easily find the main part and concentrate on it, but nobody tells CNN to do so. We think images with big size of environment information could simply add CNN difficulty to recognize, and so the key is to find the good proportion of the main part and background part. However, we also did not have time to do experiments about this idea before, though the correctness of this idea supports the base of this section and the algorithm. Lets continue this section. As we mentioned before, we finally used simply one ResNet to recognize the action, which achieves a better balance of accuracy and calculation cost. But we think we should also introduce the net we designed to compete with ResNet. Whats more, we also hope we could improve the temporal net in the future to achieve much higher merging accuracy with low cost. Inspired by the famous two-stream action recognition model, we also designed two nets to analyze the spatial and temporal information respectively. The analyzation results of these two nets will be merged to get the final recognition score for each sequence. The merging coefficient is based on the training loss of the two nets in the last epoch, the higher the training loss, the lower the merging coefficient. The structure of the one-person action recognition algorithm is given below:

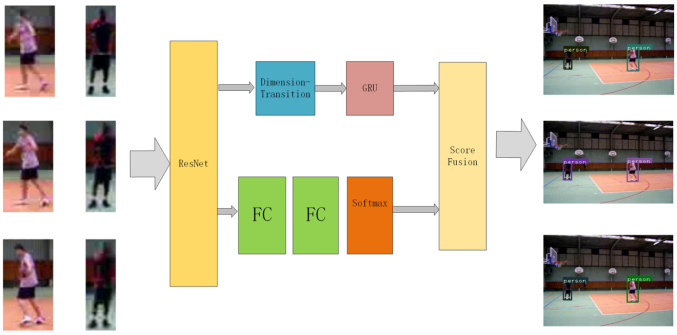

Fig. 1. Framework of individual action recognition

Spatial Net The spatial net uses ResNet to analyze the appearance information. We used the weight of ResNet pre-trained on ImageNet to begin, and we also made no changes to it. Since many actions could also be recognized from appearance, the ResNet alone could generate good results even without the help of the temporal net. As one of the two-stream network, the results of the ResNet in all the frames of a sequence will be combined to produce class scores of all the action finally.

Temporal Net The simplest temporal net may be one linear summation of the spatial nets output. Such a method, though very simple and also cannot generate more useful temporal information, works well actually. However, we also hope to get more analyzation on the temporal sequence, and so one of the widely-used RNN network, GRU, is considered there to analyze the sequences temporal information. The temporal net is built based on the spatial net: the final feature layer of the spatial net gives the input matrix to the GRU network. Since we want an algorithm with low computation cost, the calculation results of the final layer of different frames are resized and connected into a series of two-dimension matrices, which are used as the input of the GRU network. After accepting all the frames of a given sequence, the GRU will also generate the class scores of all the action directly. Apart from GRU network, we also considered the factorization of C3D net. The original C3D is abandoned since the big difficulty to train, but the factorization result: C2D+1D, is much easier. We also take the final layer of spatial net out and resize each final-layer of one frame into one 1-D vector, and then they could be connected along the temporal dimension and get one final 2-D matrix to represent one short sequence. Then we designed one simple CNN to train on it and get the prediction of such a temporal network. We also want to re-mention the reason that we do not use the optical flow image there. First, the dense optical flow cannot give the temporal analyzation result but only another new view of the spatial information. Secondly, the dense optical flow net requires the pre-calculation result of many dense-optical-flow images with different interval, which has very big calculation cost and cannot save calculation using simple ways like sample-interpolation. As such, we did not try it since it is not coordinate with the main propose of this paper-for a faster algorithm.

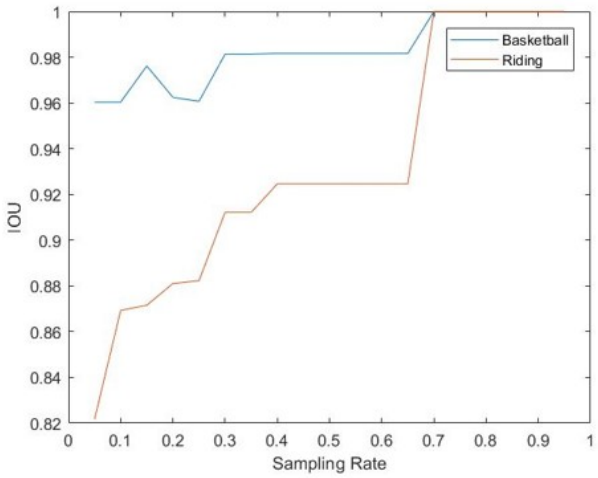

Fig. 2. The effect of sampling rate on IOU

Training There are several stages of the training process to improve the total algorithm performance. Since the ResNet alone could achieve many recognition tasks, we hope to make full use of it, and the two nets could each plays their specific role in the recognition process. Firstly, we trained the ResNet alone for a few epochs. Since the GRU net takes the results of ResNet as input, if the ResNet cannot work well, the training of GRU has no meaning at all. The second stage is prepared for GRU network alone. After a few epochs, the GRU could also achieve some easy recognition tasks. We do not change the ResNet in this stage because we hope GRU could converge faster. The third stage is to train these two networks together. Since the GRU is connected on the top of ResNet, the loss of GRU could also reach all the layers of ResNet by back-propagation. In this situation, the loss error of GRU will be combined with the merging coefficient to back-propagate in the ResNet. Currently, Alternative training has not been tested.

IV. Experiments

In this section, we implemented the proposed algorithm, and shows the result of it. We mainly used the famous action recognition dataset - UCF101 with individual annotations to test our ideas. A few videos without annotations or whose labelled frame less than 8 are excluded. And since not all person in UCF101 are labeled, so we only test the labeled person for action recognition. Since the limited experimental equipment and the time we could use them, we only chose 9 kinds of sports video randomly from UCF101s first split.

A. Person detection

Since YOLO itself could already achieve excellent performance in object detection, there is no need to show it again there, and we mainly do experiments based on the different changes. The figure below shows the bounding box IOU between the 100% sampling rate and those calculated by sampling-interpolation with other sampling rate.

Table 4.1 Influence of sampling rate on speed

	0.3 s^{-1}	0.5 s^{-1}	0.6 s^{-1}
Non-sampling	0.023/img	0.023/img	0.023/img
sampling	0.0073/img	0.0119/img	0.0141/img

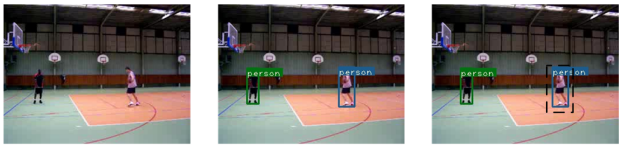

Fig. 3. The effect of expanded bounding box

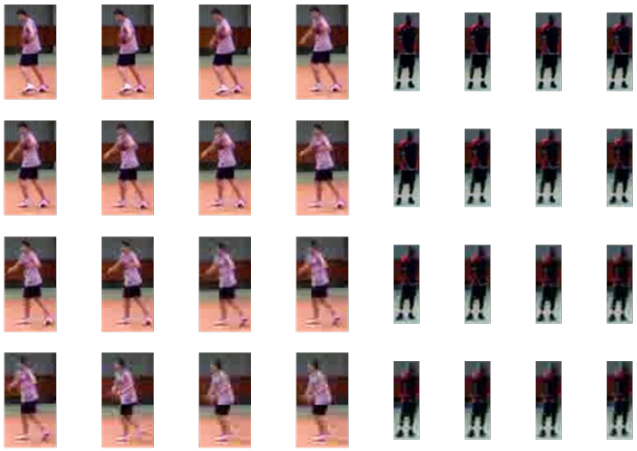

Fig. 4. Results of sequence connection

It could be seen that a sampling rate of 0.7 could already achieve especially high positioning accuracy, which could save about 30% calculation, and a sampling rate of 0.3 could achieve a good balance between positioning accuracy and total calculation. Such idea could help the proposed algorithm speed up a lot, and it could also be considered in other work.

The chart below shows the relationship between sampling rate and average human detection speed per frame: The figure below shows the idea of expanded bounding boxes. The expanded bounding boxes could take more environmental information into consideration and improve the effect of final detection.

The other experiments, like the comparison of the improved criteria and the original YOLO3, the IOU of our YOLO method with the true bounding box, cannot be finished since we do not have enough time. However, from what we tested in a small scale, the results are satisfied and we think such method is enough to support the following steps. We will add them in the future.

B. Sequence Connection

Following the sequence connection algorithm mentioned before, the figure below shows its effect:

C. Action Recognition of Person-level Sequence

This is the individual test of action recognition network, which means the bounding boxes are not acquired from YOLO network mentioned before, but from direct bounding boxes coordinates labelled by person. In the following experiments, we found most figures of UCF101 has especially low resolution for individual person, and so most current object detection algorithm even cannot detect the moving person, and so they cannot provide strong support for action recognition network for now. We already finished the total programming work now, but our laptop cannot provide end-to-end training and testing. The results are shown below:

Another work we did during my undergraduate years are shown there to be used as a comparison between different action recognition network:

Table 4.2: Comparison of individual action recognition between different algorithms on UCF09

	Training Iterations	Accuracy	Test Speed (second/frame)	Final Speed (s/f)	Test Usage
Our Algorithm	14	92.5%	Generate Proposal: 0.0119 Action Recognition: 0.0053	**0.0172**	**Multi-Person Action Recognition**
Two-stream ResNet	12	95%	One-Stream: 0.0042 Optical Flow Generation: 0.0640	0.0726	Scene Classification
C3D	Fail Tried 150 iteration	——	0.0041	0.0041	Scene Classification

Our results show the strong power of CNN again, especially when comparing with its combinations. However, it may not be the only contribution to CNNs strong competition. Since the limit of available memory, we only extracted the final layer of CNN to connect to the following temporal net. Since the last layer is already very abstract representation of the input image, it may not contain too much useful temporal information. Another reason may be the incompatibility of the CNN and temporal network. The two net has different mechanism, and they are trained to achieve different purposes. But we didnt have time to improve the training process, like training the two net together. We believe the error of the temporal networks back-propagation to the CNN will also improve the temporal nets performance, but we didnt have time to try it.

Table 4.3 Experimental Result of UCF24

	Top1 (%)	Top5 (%)	Batch Time (s)
CNN	82.031	95.573	0.0315
CNN+GRU	80.078	95.703	0.0435
CNN+CNN (C2D+C1D)	82.292	97.005	0.0417

V. Conclusions

We proposed a new algorithm to achieve multi-person action recognition with fast speed. The main innovation is about the picture-transmission mechanism with weighted sampling mechanism, which brings more useful information comparing with feature-transmission. We also proposed the sampling-interpolation to help speed it up. The action recognition part used one CNN net and one GRU net, which from our experiments achieve the best balance of accuracy and speed. The main problem of the algorithm is that we still have many ideas to test, and these work would become the future work, if we have chance to continue. Finally, this paper is simply one medium-summary about the short-term project, after all, but we still hope we could continue it in the future to make it a higher-level academic fruit.

References

Please number citations consecutively within brackets [1]. The sentence punctuation follows the bracket [2]. Refer simply to the reference number, as in [3]—do not use "Ref. [3]" or "reference [3]" except at the beginning of a sentence: "Reference [3] was the first ..."

Number footnotes separately in superscripts. Place the actual footnote at the bottom of the column in which it was cited. Do not put footnotes in the abstract or reference list. Use letters for table footnotes.

Unless there are six authors or more give all authors' names; do not use "et al.". Papers that have not been published, even if they have been submitted for publication, should be cited as "unpublished" [4]. Papers that have been accepted for publication should be cited as "in press" [5]. Capitalize only the first word in a paper title, except for proper nouns and element symbols.

For papers published in translation journals, please give the English citation first, followed by the original foreign-language citation [6].

References

[1] Girshick R. Fast R-CNN[J]. Computer Science, 2015.

[2] Simonyan K, Zisserman A. Two-stream convolutional networks for action recognition in videos[J]. 2014, 1(4):568-576.

[3] Deng J, Dong W, Socher R, et al. ImageNet: A large-scale hierarchical image database[C]// Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on. IEEE, 2009:248-255.

[4] Krizhevsky A, Sutskever I, Hinton G E. ImageNet classification with deep convolutional neural networks[C]// International Conference on Neural Information Processing Systems. Curran Associates Inc. 2012:1097-1105.

[5] Szegedy C, Liu W, Jia Y, et al. Going deeper with convolutions[J]. 2014:1-9.

[6] Simonyan K, Zisserman A. Very Deep Convolutional Networks for Large-Scale Image Recognition[J]. Computer Science, 2014.

[7] He K, Zhang X, Ren S, et al. Deep Residual Learning for Image Recognition[J]. 2015:770-778.

[8] Ren S, He K, Girshick R, et al. Faster R-CNN: towards real-time object detection with region proposal networks[C]// International Conference on Neural Information Processing Systems. MIT Press, 2015:91-99

[9] Redmon J, Divvala S, Girshick R, et al. You Only Look Once: Unified, Real-Time Object Detection[C]// Computer Vision and Pattern Recognition. IEEE, 2016:779-788.

[10] Girshick R, Donahue J, Darrell T, et al. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation[C]// IEEE Conference on Computer Vision and Pattern Recognition. IEEE Computer Society, 2014:580-587.

[11] Girshick R. Fast R-CNN[J]. Computer Science, 2015.

[12] Redmon J, Farhadi A. YOLO9000: Better, Faster, Stronger[J]. 2016:6517-6525.

[13] Redmon J, Farhadi A. YOLOv3: An Incremental Improvement[J]. 2018.

[14] Lin T, Zhao X, Shou Z. Temporal Convolution Based Action Proposal: Submission to ActivityNet 2017[J]. 2017.

[15] Singh G, Saha S, Sapienza M, et al. Online Real-Time Multiple Spatiotemporal Action Localisation and Prediction[J]. 2016.

[16] Fernando B, Gould S. Learning End-to-end Video Classification with Rank-Pooling[C]// ICML. 2016.

[17] Sevillalara L, Liao Y, Guney F, et al. On the Integration of Optical Flow and Action Recognition[J]. 2017.

[18] Du T, Wang H, Torresani L, et al. A Closer Look at Spatiotemporal Convolutions for Action Recognition[J]. 2017.

[19] Carreira J, Zisserman A. Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset[J]. 2017:4724-4733.

[20] Qiu Z, Yao T, Mei T. Learning Spatio-Temporal Representation with Pseudo-3D Residual Networks[C]// IEEE International Conference on Computer Vision. IEEE Computer Society, 2017:5534-5542.

[21] Sun L, Jia K, Yeung D Y, et al. Human Action Recognition Using Factorized Spatio-Temporal Convolutional Networks[C]// IEEE International Conference on Computer Vision. IEEE, 2015:4597-4605.

[22] Bagautdinov T, Alahi A, Fleuret F, et al. Social Scene Understanding: End-to-End Multi-person Action Localization and Collective Activity Recognition[C]// IEEE Conference on Computer Vision and Pattern Recognition. IEEE Computer Society, 2017:3425-3434.

[23] Saha S, Singh G, Sapienza M, et al. Deep Learning for Detecting Multiple Space-Time Action Tubes in Videos[J]. 2016.

[24] Kalogeiton V, Weinzaepfel P, Ferrari V, et al. Action Tubelet Detector for Spatio-Temporal Action Localization[C]// IEEE International Conference on Computer Vision. IEEE, 2017:4415-4423.

[25] Chen C. Tube Convolutional Neural Network (T-CNN) for Action Detection in Videos[C]// IEEE International Conference on Computer Vision. IEEE Computer Society, 2017:5823-5832.

[26] Saha S, Singh G, Sapienza M, et al. Deep Learning for Detecting Multiple Space-Time Action Tubes in Videos[J]. 2016.

[27] Ji S, Xu W, Yang M, et al. 3D Convolutional Neural Networks for Human Action Recognition[J]. IEEE Transactions on Pattern Analysis Machine Intelligence, 2012, 35(1):221-231.

[28] Ji S, Xu W, Yang M, et al. 3D Convolutional Neural Networks for Human Action Recognition[C]// International Conference on Machine Learning. DBLP, 2010:495-502.

[29] Peng X, Schmid C. Multi-region Two-Stream R-CNN for Action Detection[C]// European Conference on Computer Vision. Springer International Publishing, 2016:744-759.

[30] Zhu W, Lan C, Xing J, et al. Co-occurrence Feature Learning for Skeleton based Action Recognition using Regularized Deep LSTM Networks[J]. ł, 2016:3697-3703.

[31] Chen W, Corso J J. Action Detection by Implicit Intentional Motion Clustering[C]// IEEE International Conference on Computer Vision. IEEE Computer Society, 2015:3298-3306.

[32] Gemert J C V, Jain M, Gati E, et al. APT: Action localization Proposals from dense Trajectories[C]// BMVC. 2015.

[33] Puscas M M, Sangineto E, Culibrk D, et al. Unsupervised Tube Extraction Using Transductive Learning and Dense Trajectories[C]// IEEE International Conference on Computer Vision. IEEE, 2015:1653-1661.

[34] Mikolov T , Sutskever I , Chen K , et al. Distributed Representations of Words and Phrases and their Compositionality[J]. Advances in Neural Information Processing Systems, 2013, 26:3111-3119.