DIVERSITY ENCOURAGING ENSEMBLE OF CONVOLUTIONAL NETWORKS FOR HIGH PERFORMANCE ACTION RECOGNITION

Hao Yang, Chunfeng Yuan, Junliang Xing, Weiming Hu*

CAS Center for Excellence in Brain Science and Intelligence Technology;
National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences;
University of Chinese Academy of Sciences, Beijing, China
{hao.yang, cfyuan, jlxing, wmhu}@nlpr.ia.ac.cn

ABSTRACT

We present a simple and effective ensemble method, Diversity Encouraging Ensemble (DEE), for deep convolutional networks to boost their performances. By training the convolutional network in two stages, we generate multiple component networks without adding any training cost. On the one hand, we modify the structure parameters of component networks in the training process to enlarge the diversities of the networks, which is found to be beneficial to improving the ensemble performance. On the other hand, we exploit monotonous decreasing learning rate schedule to accelerate the speed of deep network converging to different local minima, and we decrease the training time of integrating multiple networks to that of training a single network from traditional multi-step learning policy. We evaluate our ensemble method on two challenging action datasets, UCF-101 and HMDB-51, and obtain performance improvements from single deep network and other ensemble methods. Our results also outperform many state-of-the-art action recognition methods.

Index Terms— Diversity Encouraging Ensemble, Convolutional Neural Network, Action Recognition

1. INTRODUCTION

In recent years, deep learning based models have become the dominant approaches in many research fields, such as computer vision [1], machine translation [2] and speech recognition [3]. An effective and common practice to improve the performance of deep networks is to train an ensemble of multiple networks. The ensemble methods can be categorized as implicit ensemble and explicit ensemble. Implicit ensemble methods [4, 5, 6, 7, 8] train a single neural network with lots of branches or paths. During test, the results from all the branches or paths are fused together to generate the final results. Explicit ensemble methods [9, 10, 11, 12, 13] individually train multiple networks with different initialization, and learn a weight for each component network based on their classification precision on the validation set [11, 12]. Then the final predictions are the average [9, 10] or weighted average [11, 12] of these component networks.

It is wildly accepted that the diversity is an important property of ensemble methods [14]. The Diversity-Penalizing Ensemble [13] trains multiple deep networks individually and minimizes the KL divergence loss to impel each network to approximate the ensemble prediction. During test, only one network is used to perform predicting, which decreases the test time. The training time, however, is very expensive since multiple networks need to be independently trained from scratch. Moreover, impelling all the networks to approximate the same ensemble prediction may reduce the diversities of component networks. The Snapshot Ensemble [15] uses periodically restarting learning rate proposed in [16] to converge and escape from local minima to reduce the training time. Also, if the diversities between these local minima are very small, it may fail to skip from these minima and result in unsatisfactory ensemble performance.

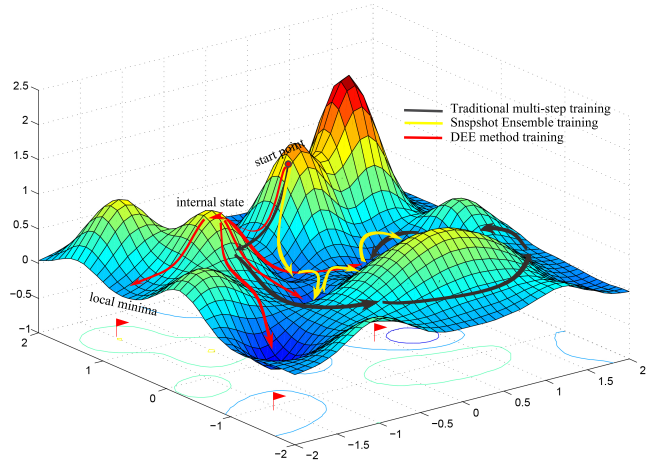

Fig. 1: The illustration of different optimization schedules. **Black**: the traditional multi-step learning rate optimization schedule; **Yellow**: the Snapshot Ensemble optimization schedule; **Red**: our DEE optimization schedule. The distances between these local minima of our ensemble method are much larger than those of the traditional method and the Snapshot Ensemble.

In this paper, we propose the Diversity Encouraging Ensemble method, denoted as DEE, which trains an ensemble of multiple convolutional networks of large diversities without any additional training cost. Specifically, we first train a deep convolutional network with large learning rate for a few epochs to an internal state. Then, we modify the structure parameters of the network and fine-tune the weights from internal state to enlarge the diversities of local component net-

*Corresponding Author

works. The high-level overview of our ensemble is illustrated in Fig. 1. Our ensemble method converges to multiple local minima and the distances between these minima are much larger. Meanwhile, we reuse the internal state to let it converge to multiple local minima, which reduces the training time dramatically. To further decrease the training time, we train each network with a monotonous decreasing learning rate, which helps the model converging much faster.

We apply the proposed DEE method to the task of action recognition, as it has been an common practice to boost action recognition performance from ensemble [17, 18, 19, 20, 21, 22]. For example, the ensemble of SpatialNet and TemporalNet [17, 18] greatly improves the performance of action recognition. Temporal Segment Network (TSN) [19] is an ensemble of six deep convolutional networks, *i.e.* three SpatialNets and three TemporalNets, and achieves the current state-of-the-art performance. Ensemble of deep features and handcrafted features (such as IDTs [23]) is also an effective method [20, 21, 22] to improve the action classification performance. Compared to these methods, our DEE method obtains best performance from simple diversity encouraging ensemble, without using carefully designed model architectures and/or tuned learning strategies.

To summarize, in this paper we have made the following three main contributions:

- We propose a new ensemble method, Diversity Encouraging Ensemble, which modifies the structure parameters of the networks in the training process to enlarge the diversities of the component networks.
- We reuse the internal training state of a deep convolutional model and exploit the monotonous decreasing learning rate to reduce the training time, which makes the DEE method very efficient.
- We apply our ensemble method for action recognition, which achieves the state-of-the-art performance on two of the most challenging action recognition datasets, UCF-101 and HMDB-51.

2. PROPOSED ENSEMBLE METHOD

Traditional network ensemble methods either have high training cost [9, 10, 11] or small diversities of the component networks [15]. The DEE method is motivated to overcome the two problems. It divides the training process into two phases, general training and specific training. In the general training phase, we train the deep network using a large constant learning rate and high dropout rate for a few epochs, so it is not prone to converging to a local minima and stop at a internal state. In this phase, the deep network learns some general features from videos, such as appearance and motions. In the specific training phase, on the one hand, we use variate structure parameters for component networks to enlarge the diversities between these models. On the other hand, we reuse the internal state and exploit the monotonous decreasing learning rate to accelerate these models converging to the local minima and reduce the training time.

2.1. Diversity Encouraging Ensemble

The traditional ensemble methods [9, 10, 11] train M component networks from random initializations. The weights are initialized from the same distribution, so the initial states of the M networks are very close. Moreover, since the M component networks are trained on the same data from the similar initial state, the networks are thus prone to converging to the same or very close local minima. The lack of diversity is found to hurt the final ensemble performance [14].

To address this problem, we encourage diversities of the component networks in the two training phases. In the general training phase, we train the deep network with a high dropout rate to learn general features from actions. In the specific training phase, we modify the structure parameters from internal model by using different dropout rates for the dropout layers. The use of different dropout parameters in the two training phases as well as different dropout rates for the component networks encourage large diversities between component networks, which benefits the ensemble performance.

Here we elaborate on why using different dropout rate for each component network is diversity encouraged. Given a dropout layer with dropout rate r, the layer will randomly set some activations from last fully-connected layer to 0 with proportion of r. Modifying the dropout rate changes the combination of activations from last fully-connected layer to next layer, which compels the fully-connected layers to adapt their weights significantly to match the new combination of activations. So the dropout modification changes the network weights significantly and it increases the diversities of the component networks.

2.2. Efficient Ensemble Learning

Most of state-of-the-art deep convolutional neural networks [6, 19, 24, 25, 26] are trained by the Stochastic Gradient Decent (SGD) with momentum. The training objective function is denoted as $f : R^n \rightarrow R$, and $w_t \in R^n$ is the n learning weights. The momentum is denoted as m_t and v_t is the velocity vector at the t^{th} iteration, which is initialized as $\mathbf{0}$. The weights w_t are updated as follows:

$$v_{t+1} = m_t v_t - \eta_t \nabla f_t(x), \tag{1}$$

$$w_{t+1} = w_t + v_{t+1}, \tag{2}$$

where $\nabla f_t(x)$ is the gradient of the objective function respect to the weights, and η_t is the learning rate, which is a constant and is divided by a constant periodically.

In order to accelerate the convergence and decrease the training time of the M component networks, we propose to employ the schedule of the monotonous decreasing learning rate. In the specific training phase, we decrease the learning rate monotonously from initial value η_0 to the minimum η_{min}, instead of dividing learning rate periodically as traditional multi-step learning policy. Let e_{num} denote the total

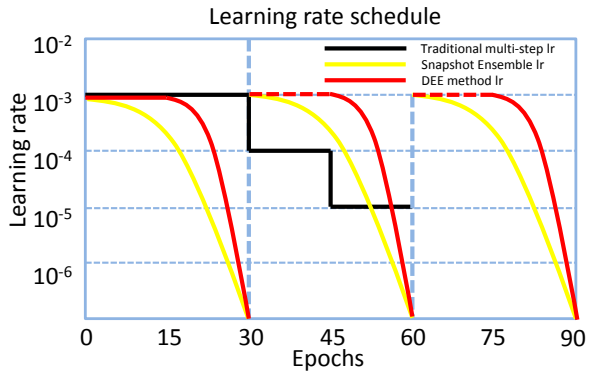

Fig. 2: The training process of SpatialNet [18] by different learning rate schedule of traditional multi-step learning policy, the Snapshot Ensemble and DEE method.

number of epochs and e_{cur} denote the current trained epoch number. The learning rate at current epoch is formulated as:

$$\eta_t = \eta_{min} + \frac{1}{2}(\eta_0 - \eta_{min})[1 + \cos(\frac{e_{cur}\pi}{e_{num}})]. \quad (3)$$

In all of our experiments, we set $\eta_{min} = 0$. So we simplify the function as:

$$\eta_t = \frac{\eta_0}{2}[1 + \cos(\frac{e_{cur}\pi}{e_{num}})], \quad (4)$$

and we set $e_{num} = 15$ in all of our experiments for each component network.

In Fig. 2, we show learning rate schedules of finetuning the deeper SpatialNet [18] from VGG-16 [25] which is pre-trained on ImageNet and we compare the training time of single deep network trained by traditional multi-step learning policy, the Snapshot Ensemble [15] with three local minima, and the DEE method with three component networks. The total training time of traditional multi-step learning policy is 60 epochs, which is shown as black line. The Snapshot Ensemble periodically restart learning rate to converge to three local minima with 30 epochs for each period as shown in yellow line. The DEE method, shown as red line, trains three component networks in two phases, with 15 epochs for each phase. We reuse the internal model trained in the general training phase, which decreases the training time by $1/3$, illustrated as red dash lines. The total training time of our DEE method is identical to that of a single network trained by traditional multi-step learning policy and is much less than that of the Snapshot Ensemble.

2.3. Ensemble in the Test Stage

After the two training phases, we obtain M component models, $\{p_m, m = 1, ..., M\}$. Given a test example x, let $p_m(x)$ represent the prediction distribution from Softmax layer of the m^{th} network and $p_{en}(x)$ denote the ensemble prediction. We design two fusion methods to compute the ensemble of M networks. The first one simply averages the M predictions:

$$p_{en} = \frac{1}{M}\sum_{m=1}^{M} p_m(x). \quad (5)$$

The other bases on the information entropy, which evaluates the average indeterminacy of a random distribution. Let $p_{mi}(x), i = 1, ..., C$ represent the Softmax scores of the m^{th} network. The information entropy is formulated as:

$$H[p_m(x)] = -\sum_{i=1}^{C} p_{mi}(x)\log p_{mi}(x). \quad (6)$$

When p_m is reliable, it is usually sparse with low entropy of the distribution, *i.e.* only a few entries of p_m have large values, while other entries are small or approaching zeros; conversely, when p_m is not reliable, its entry values (class probabilities) tend to spread evenly over all the action categories. So we propose a entropy-based weighted averaging ensemble of M networks as:

$$p_{en} = \sum_{m=1}^{M} \alpha_m p_m(x), \quad (7)$$

where $\alpha_m = \frac{1}{H[p_m(x)]}$ represents the confidence of each component network.

3. EXPERIMENTS

3.1. Experimental Settings

The UCF-101 [27] is a dataset of realistic action videos, containing 101 action categories with 13320 videos. We report the average accuracy of standard splits from [27]. The HMDB-51 dataset [28] contains 6849 clips divided into 51 action categories. We use the splits provided by [28] of raw videos without stabilization.

The DEE model consists of six component networks, three SpatialNets and three TemporalNets, whose structure are all same as VGG-16 [25] excepting the dropout layer and first convolutional layer. We set the dropout rate as (0.9, 0.9) for the two dropout layers in general training phase and we modify them to (0.7, 0.7), (0.6, 0.7), (0.7, 0.8) for the three networks in the specific training phase respectively. Please referent [18] for the details of SpatialNet and TemporalNet. We train the deep networks by mini-batch Stochastic Gradient Decent (SGD) with momentum. We set the momentum as 0.9 and weight-decay as 0.0005. We train the SpatialNets with batch-size as 16 and initial learning rate as 0.001. The TemporalNets are trained by setting batch-size as 22 and initial learning rate as 0.003. Data augmentation techniques are used, such as corner cropping, multi-scale cropping and random flipping, to avoid over-fitting. In test, we randomly sample 25 clips from each video. Then the standard 10-views of cropping and flipping are applied. The prediction score of each video is the average of all the samples.

3.2. Comparison with Different Ensemble Methods

In the first experiment, we compare the performance of our DEE method with the Snapshot Ensemble [15] method on the UCF-101 dataset. the Snapshot Ensemble is originally used to classify images and we extend it to classify actions by exploiting cyclical cosine annealing. We train the Snapshot Ensemble networks on RGB and optical flow respectively. And

Table 1: Evaluating the diversity of the DEE method on UCF-101 split1.

Models	SpatialNets					TemporalNets				
	p1	p2	p3	AVG	EnAVG	p1	p2	p3	AVG	EnAVG
Deeper Two-stream [18]	79.8					85.7				
Snapshot Ensemble [15]	**80.76**	80.54	**80.20**	81.10	82.08	84.83	**85.86**	85.41	85.77	86.52
Proposed DEE method	80.65	**80.91**	80.12	**81.68**	**83.56**	**84.88**	85.59	**85.43**	**85.86**	**87.34**

each training path has three periods with 30 epochs for each period. It use the same batch-size and initial learning rate as the DEE method. We report the component accuracies and two ensemble results of our ensemble method and the Snapshot Ensemble in Table 1. The AVG represents the simple average ensemble and the EnAVG represents the entropy-based weighted average ensemble.

From Table 1, we have four conclusions. 1) For the SpatialNets, each component network accuracy of the DEE method outperforms the baseline performance [18]. It illustrates that the proposed two phases training schedule with monotonous decreasing learning policy not only accelerates the training but also improves the performance of action recognition. 2) From the last row, the AVG and EnAVG accuracies of the DEE method significantly outperform every component network without any additional training cost. 3) For the TemporalNets, the AVG accuracy of the Snapshot Ensemble is lower than the single component performance of $p2$, which indicates the complement between these local minima of the Snapshot Ensemble is very poor. 4) In the last two rows, we compare the diversities of the Snapshot Ensemble and the DEE methods. The components performance of the two ensemble methods are roughly equal but our ensemble accuracies outperform the Snapshot Ensemble. It illustrates the diversities of our component networks are larger than that of the Snapshot Ensemble.

In the second experiment, we evaluate the ensemble of SpatialNets and TemporalNets on UCF-101 and HMDB-51 datasets. We report the EnAVG results of the DEE method and Snapshot Ensemble methods. The Deeper Two-stream [18] only fuses two networks, while the others fuse six networks, three SpatialNets and three TemporalNets. As shown in Fig. 3, the DEE method outperforms the Snapshot Ensemble [15]. Also, the DEE method is comparable with currently best model TSN (2 modilities) [19] on the UCF-101 dataset and outperforms it on the HMDB-51 dataset.

3.3. Comparison with Action Recognition Methods

In Table 2, we list the results of current state-of-the-art methods for action recognition on UCF-101 and HMDB-51 benchmarks. The DEE method outperforms the traditional action recognition methods, such as improved dense trajectories (IDTs) [23] and IDTs coding with fisher vector [29]. The DEE method also outperforms other competitive deep learning based action classification methods, such as the Two-stream [17], the deeper Two-stream [18], the fusion Two-stream [21], and the factorized spatio-temporal convolutional networks (FstCN) [30] on both datasets. In order to improve the ensemble performance further, we use the ensemble of nine networks and each three networks are trained on RGB frame, stacking flow filed and warped flow filed as TSN (3 modalities) [31]. We get 94.3% and 69.7% on the UCF-101 and HMDB-51 datasets respectively and it outperforms TSN (3 modalities) [31] on both datasets.

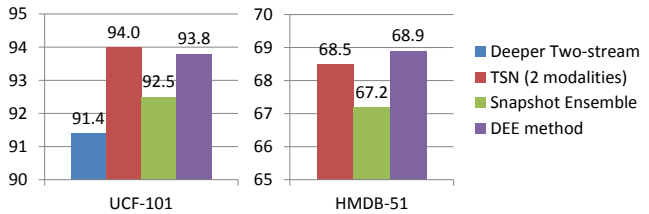

Fig. 3: The accuracy of the ensemble of SpatialNets and TemporalNets on the UCF-101 and HMDB-51 datasets.

Table 2: Comparing with current state-of-the-art methods.

UCF-101		HMDB-51	
Wang et al. [23]	85.9	Wang et al. [23]	57.2
Peng et al. [29]	87.9	Peng et al. [29]	61.1
Wang et al. [20]	90.3	Wang et al. [20]	63.2
Simonyan et al. [17]	86.9	Simonyan et al. [17]	58.0
Sun et al. [30]	87.9	Sun et al. [30]	58.6
Wang et al. [18]	91.4	Wang et al. [31]	63.4
Zhu et al. [32]	93.1	Zhu et al. [32]	63.3
Wang et al. [19]	94.2	Wang et al. [19]	69.4
Christoph et al. [21]	91.8	Varol et al. [33]	64.8
DEE method (6 Nets)	93.8	DEE method (6 Nets)	68.9
DEE method (9 Nets)	**94.3**	DEE method (9 Nets)	**69.7**

4. CONCLUSION

In this paper, we have propose Diversity Encouraging Ensemble, an effective and efficient ensemble method, which produces the trained networks with large diversity, which is beneficial to final ensemble performance. The proposed ensemble method reuse the internal state and exploit monotonous decreasing learning policy to greatly reduce the training time of the component networks. We have applied the proposed ensemble method on the task of action recognition and it demonstrates very promising performance compared to many competitive methods. In future work, we plan to further improve the test time of the proposed ensemble method.

5. ACKNOWLEDGEMENT

This work is partly supported by the 973 basic research program of China (Grant No. 2014CB349303), the Natural Science Foundation of China (Grant No. U1636218, 61472420, 61472063, 61370185, 61472421, 61672519), the Strategic Priority Research Program of the CAS (Grant No. XDB02070003), and the CAS External cooperation key project.

6. REFERENCES

[1] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, 2012.

[2] Ilya Sutskever, Oriol Vinyals, and Quoc Le, "Sequence to sequence learning with neural networks," in *Advances in Neural Information Processing Systems*, 2014.

[3] Alex Graves and Navdeep Jaitly, "Towards end-to-end speech recognition with recurrent neural networks.," in *International Conference on Machine Learning*, 2014.

[4] Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting.," *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[5] Li Wan, Matthew Zeiler, Sixin Zhang, Yann L Cun, and Rob Fergus, "Regularization of neural networks using dropconnect," in *International Conference on Machine Learning*, 2013.

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," *arXiv preprint arXiv:1512.03385*, 2015.

[7] Andreas Veit, Michael Wilber, and Serge Belongie, "Residual networks behave like ensembles of relatively shallow networks," *arXiv preprint arXiv:1605.06431v2*, 2016.

[8] Gao Huang, Yu Sun, Zhuang Liu, Daniel Sedra, and Kilian Weinberger, "Deep networks with stochastic depth," *arXiv preprint arXiv:1603.09382*, 2016.

[9] Liran Chen, "Learning ensembles of convolutional neural networks," 2014.

[10] Thomas G Dietterich, "Ensemble methods in machine learning," in *International Workshop on Multiple Classifier Systems*, 2000.

[11] ZongYuan Ge, Alex Bewley, Christopher McCool, Peter Corke, Ben Upcroft, and Conrad Sanderson, "Fine-grained glassification via mixture of deep convolutional neural networks," in *IEEE Winter Conference on Applications of Computer Vision*, 2016.

[12] Xavier Frazao and Luís A Alexandre, "Weighted convolutional neural network ensemble," in *Iberoamerican Congress on Pattern Recognition*, 2014.

[13] Xiaohui Zhang, Daniel Povey, and Sanjeev Khudanpur, "A diversity-penalizing ensemble training method for deep learning," in *Annual Conference of International Speech Communication Association*, 2015.

[14] Zhihua Zhou, *Ensemble Methods: Foundations and Algorithms*, Chapman and Hall/CRC, 1st edition, 2012.

[15] Gao Huang, Yixuan Li, and Geoff Pleiss, "Snapshot Ensembles: Train 1, get m for free," in *Under review at ICLR*, 2017.

[16] Ilya Loshchilov and Frank Hutter, "SGDR: Stochastic gradient descent with restarts," *arXiv preprint arXiv:1608.03983*, 2016.

[17] Karen Simonyan and Andrew Zisserman, "Two-stream convolutional networks for action recognition in videos," in *Advances in Neural Information Processing Systems*, 2014.

[18] Limin Wang, Yuanjun Xiong, Zhe Wang, and Yu Qiao, "Towards good practices for very deep two-stream convnets," *arXiv preprint arXiv:1507.02159*, 2015.

[19] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool, "Temporal Segment Networks: Towards good practices for deep action recognition," in *European Conference on Computer Vision*. Springer, 2016.

[20] Limin Wang, Yu Qiao, and Xiaoou Tang, "Action recognition with trajectory-pooled deep-convolutional descriptors," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[21] Christoph Feichtenhofer, Axel Pinz, and Andrew Zisserman, "Convolutional two-stream network fusion for video action recognition," *arXiv preprint arXiv:1604.06573*, 2016.

[22] Zhenyang Li, Efstratios Gavves, Mihir Jain, and Cees GM Snoek, "VideoLSTM convolves, attends and flows for action recognition," *arXiv preprint arXiv:1607.01794*, 2016.

[23] Heng Wang and Cordelia Schmid, "Action recognition with improved trajectories," in *IEEE International Conference on Computer Vision*, 2013.

[24] Tran Du, Lubomir Bourdev, Rob Fergus, and Lorenzo Torresani, "Learning spatiotemporal features with 3d convolutional networks," in *IEEE International Conference on Computer Vision*, 2015.

[25] Karen Simonyan and Andrew Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[26] Ross Girshick, "Fast R-CNN," in *IEEE International Conference on Computer Vision*, 2015.

[27] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah, "UCF101: A dataset of 101 human actions classes from videos in the wild," *arXiv preprint arXiv:1212.0402*, 2012.

[28] Hildegard Kuehne, Hueihan Jhuang, Estíbaliz Garrote, Tomaso Poggio, and Thomas Serre, "HMDB: A large video database for human motion recognition," in *International Conference on Computer Vision*. IEEE, 2011.

[29] Xiaojiang Peng, Limin Wang, Xingxing Wang, and Yu Qiao, "Bag of visual words and fusion methods for action recognition: Comprehensive study and good practice," *Computer Vision and Image Understanding*, vol. 150, pp. 109–125, 2016.

[30] Lin Sun, Kui Jia, Dit-Yan Yeung, and Bertram E Shi, "Human action recognition using factorized spatio-temporal convolutional networks," in *IEEE International Conference on Computer Vision*, 2015.

[31] Xiaolong Wang, Farhadi Ali, and Gupta Abhinav, "Actions transformations," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[32] Wangjiang Zhu, Jie Hu, Gang Sun, Xudong Cao, and Yu Qiao, "A key volume mining deep framework for action recognition," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[33] Gül Varol, Ivan Laptev, and Cordelia Schmid, "Long-term temporal convolutions for action recognition," *arXiv preprint arXiv:1604.04494*, 2016.