Optical Engineering

OpticalEngineering.SPIEDigitalLibrary.org

Weighing classes and streams: toward better methods for two-stream convolutional networks

Hoseong Kim
Youngjung Uh
Seunghyeon Ko
Hyeran Byun

SPIE.

Hoseong Kim, Youngjung Uh, Seunghyeon Ko, Hyeran Byun, "Weighing classes and streams: toward better methods for two-stream convolutional networks," *Opt. Eng.* **55**(5), 053108 (2016), doi: 10.1117/1.OE.55.5.053108.

Weighing classes and streams: toward better methods for two-stream convolutional networks

Hoseong Kim, Youngjung Uh, Seunghyeon Ko, and Hyeran Byun*
Yonsei University, Department of Computer Science, Seoul 120-749, Republic of Korea

Abstract. The emergence of two-stream convolutional networks has boosted the performance of action recognition by concurrently extracting appearance and motion features from videos. However, most existing approaches simply combine the features by averaging the prediction scores from each recognition stream without realizing that some classes favor greater weight for appearance than motion. We propose a fusion method of two-stream convolutional networks for action recognition by introducing objective functions of weights with two assumptions: (1) the scores from streams do not weigh the same and (2) the weights vary across different classes. We evaluate our method by extensive experiments on UCF101, HMDB51, and Hollywood2 datasets in the context of action recognition. The results show that the proposed approach outperforms the standard two-stream convolutional networks by a large margin (5.7%, 4.8%, and 3.6%) on UCF101, HMDB51, and Hollywood2 datasets, respectively. [DOI: 10.1117/1.OE.55.5.053108]

Keywords: convolutional neural network; two-stream; class-wise fusion; stream-wise fusion; action recognition; pattern recognition; neural networks; computer vision.

Paper 160129L received Jan. 28, 2016; accepted for publication Apr. 11, 2016; published online May 23, 2016.

1 Introduction

Action recognition is a fundamental research problem in computer vision that can be useful to surveillance, human–computer interaction, and video retrieval. Recently, hand-crafted features and deep-learned features have been widely investigated for action recognition. Spatiotemporal interest point[1] based on the Harris corner detector is robust to illumination change, background clutter, and video noise. Wang and Schmid[2] used improved dense trajectories with rich descriptors of histogram of gradients, histogram of optical flow, and motion boundary histogram to achieve better performance. However, these local features do not consider the large intravariation in same actions due to viewpoint change and motion speed, which may lead to limited generalization.

In recent years, deep features[3,4] learned by a convolutional neural network (ConvNet) from large-scale training datasets have been used on action recognition tasks[5–9] to achieve superior performance. Ji et al.[5] extended two-dimensional convolution technique to a three-dimensional case, which reflects temporal domain in videos. Karpathy et al.[6] investigated several fusion methods that include early, late, and slow fusion with spatiotemporal CNN. Simonyan and Zisserman[7] combined an appearance cue from the spatial stream and a motion cue from the temporal stream through two-stream convolutional networks (ConvNets). Ye et al.[8] argued that the combination of the streams affects the performance. Although all the methods based on two-stream ConvNets[7,8] are the most promising ones, they naïvely use the identical weights of each stream over distinct classes. Wang et al.[9] combined the advantages of hand-crafted features that are robust to background clutter and illumination change and deep-learned features that deliver high-level semantic information. They achieved the state-of-the-art accuracy for action recognition.

We claim that spatial stream and temporal stream should have different weight for each class. Intuitively, appearance cue will be more crucial for actions with an object (e.g., PlayingGuitar and BenchPress) than without an object (e.g., SitDown and StandUp). Motion cue will be more crucial for dynamic (i.e., out-of-place) actions that are moving from the original position than for static (i.e., in-place) actions. Therefore, such characteristics should be handled by distinct weights for appearance and motion. In this letter, we propose the class-wise and stream-wise fusion method of two-stream ConvNets for action recognition by defining two objective functions of weights with two suppositions: (1) the scores from streams do not weight the same, and (2) the weights vary across different classes. Our proposed method can be adopted to other two-stream ConvNets and generalized to multistream ConvNets. Experiments show that our class-wise and stream-wise fusion method significantly improves the performance of two-stream ConvNets on three action-recognition datasets.

2 Two-Stream Convolutional Networks

Appearance and motion are the two representative types of features that can be extracted from videos containing human actions, each of which is obtained from the spatial stream and the temporal stream, respectively. The spatial net extracts appearance feature of actions and objects from every single frame. The temporal net extracts motion feature of complex actions from stacked optical flow fields of consecutive frames.

For spatial net, we adopt the state-of-the-art deep architecture (VGG16)[4] pretrained on the ImageNet dataset[10] and fine-tune the model parameters of the architecture on UCF101,[11] HMDB51,[12] and Hollywood2[13] datasets, respectively. For temporal net, we follow a principled way of

*Address all correspondence to: Hyeran Byun, E-mail: hrbyun@yonsei.ac.kr

0091-3286/2016/$25.00 © 2016 SPIE

combining several datasets based on transfer learning[14] to avoid overfitting. For instance, we pretrain the VGG16 model on UCF101 and fine-tune it on HMDB51 or Hollywood2. This is because the available datasets to train the deep features for action recognition are still rather small. Our architecture of ConvNets comprises 16 layers including 13 convolutional layers with 3×3 convolutional windows throughout the whole net and 3 fully connected layers. The input of spatial net is an RGB image whose size is $224 \times 224 \times 3$, and the input of temporal net is 10 optical flows from 11 consecutive frames whose size is $224 \times 224 \times 20$.[7] We follow the same network[4] for both spatial and temporal net. Finally, we classify human actions by combining the softmax scores from each stream with our class-wise and stream-wise fusion method. Figure 1 shows the overall pipeline of the proposed class-wise and stream-wise two-stream ConvNets.

3 Class-Wise and Stream-Wise Fusion

Let $\mathbf{X} = \{(\mathbf{x}_i, y_i) | i = 1, \ldots, N\}$ be the training set, which contains N training samples, where $\mathbf{x}_i \in \mathbb{R}^d$ is a d-dimensional feature vector, $y_i \in \{1, \ldots, C\}$ is the ground truth label of $\mathbf{x}_i$, and C is the number of classes. We denote that $\mathbf{w} = [w_1, \ldots, w_k, \ldots, w_C]$ is the weight vector of the spatial stream for the entire class, where $w_k \in [0,1]$ is the weight of the spatial stream for the k'th class.

Unlike the existing naïve fusion method of two-stream ConvNets, the unique characteristic of each action class is considered in different weights for each class and stream. They are termed class-wise and stream-wise, respectively. In other words, it is a class-wise fusion (CF) when all the classes have different weights in each stream, i.e., $w_1 \neq w_2 \neq \cdots \neq w_C$. On the other hand, it is a stream-wise fusion when the weight of a spatial stream and the weight of a temporal stream are different from each other, but all the classes have the same weights in each stream, i.e., $\mathbf{w} \neq 1 - \mathbf{w}$, $w_1 = w_2 = \cdots = w_C$.

We define the objective function $\mathbf{Q}(\mathbf{w})$ in the form of a weighted linear function as follows:

$$\mathbf{Q}(\mathbf{w}) = \sum_{k=1}^{C} \mathbf{Q}(w_k) = \sum_{k=1}^{C} \sum_{i=1}^{N_k} \mathbf{Q}_i(w_k), \tag{1}$$

$$\mathbf{Q}_i(w_k) = -[S_{ik} w_k + T_{ik}(1 - w_k)]\gamma_i, \tag{2}$$

where $N = \sum_{k=1}^{C} N_k$, and N_k is the number of training samples for the k'th class. $\mathbf{Q}_i(w_k)$ is the subobjective function for w_k and the i'th training data. $S_{ik}, T_{ik} \in \mathbb{R}$ are the prediction scores of the spatial stream and the temporal stream such that $y_i = k$, respectively. $\gamma_i \in \{-1,1\}$ is the modified indicator function that returns 1 if $y_i = \hat{y}_i$, otherwise -1, where $\hat{y}_i \in \{1, \ldots, C\}$ is the predicted label of $\mathbf{x}_i$. We refer to this approach as the class-wise and stream-wise fusion, which is processed in order of the stream-wise first and the class-wise last.

The goal is to minimize the objective function Eq. (1). We employ the stochastic gradient descent with momentum (SGD)[15] to obtain the weight parameter w_k as follows:

$$w_k = w_k - \eta \Delta w_k, \tag{3}$$

$$\Delta w_k = \eta \nabla \mathbf{Q}_i(w_k) + \alpha \Delta w_k, \tag{4}$$

$$\nabla \mathbf{Q}_i(w_k) = -(S_{ik} - T_{ik})\gamma_i, \tag{5}$$

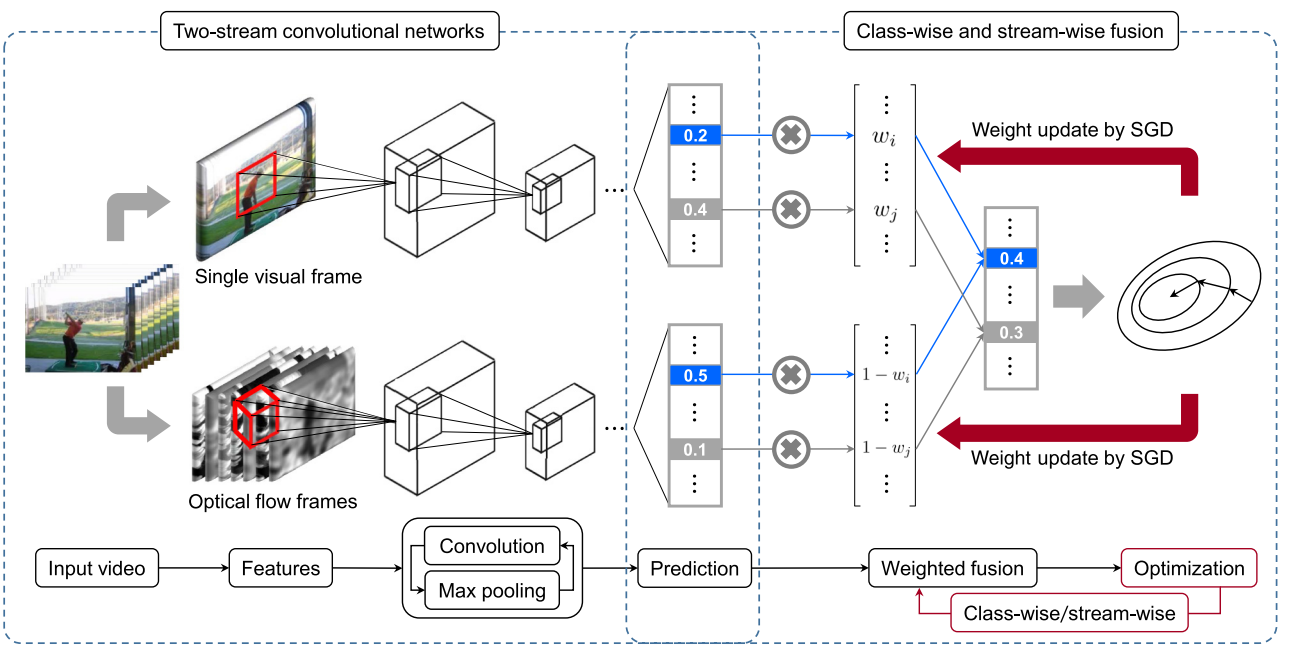

Fig. 1 The pipeline of the class-wise and stream-wise two-stream ConvNets. The whole process consists of two steps: (i) two-stream convolutional networks and (ii) class-wise and stream-wise fusion. While the red arrows and the red rectangles are processed only in a training step (e.g., weight update, optimization, and class-wise/stream-wise), the black arrows and the black rectangles are processed in both a training and a testing step. Our proposed fusion method is only performed on the fusion weights without including two-stream ConvNets. (Best viewed in color)

where η is learning rate (initially set to 5×10^{-3}), α is momentum (set to 0.9), and $\nabla \mathbf{Q}_i(w_k)$ is the gradient of the subobjective function Eq. (2) with respect to w_k.

In the class-wise and stream-wise fusion, a change of the weight in an action class may affect the accuracy of other action classes, i.e., a boost in the prediction score of a class may change the rank of the prediction score of the video in other classes. Hence, we modify the objective function Eq. (1) and the subobjective function Eq. (2) as below to compensate such side effect

$$\tilde{\mathbf{Q}}(\mathbf{w}) = \tilde{\mathbf{Q}}(w_t) = \sum_{i=1}^{N} \tilde{\mathbf{Q}}_i(w_t), \tag{6}$$

$$\tilde{\mathbf{Q}}_i(w_t) = \sum_{k=1}^{C} \mathbf{Q}_i(w_k), \tag{7}$$

where $w_t \in [0,1]$ is the weight of the spatial stream for the t'th class. The extended subobjective function $\tilde{\mathbf{Q}}_i(w_t)$ is totally unaffected by the class, i.e., $\tilde{\mathbf{Q}}_i(w_1) = \tilde{\mathbf{Q}}_i(w_t) = \tilde{\mathbf{Q}}_i(w_C)$, $\forall\ t = 1, \ldots, C$. The gradient of the subobjective function Eq. (7) with respect to w_t is computed as follows:

$$\nabla \tilde{\mathbf{Q}}_i(w_t) = \sum_{k=1}^{C} \nabla \mathbf{Q}_i(w_k). \tag{8}$$

Unlike the subobjective function Eq. (2), the extended subobjective function Eq. (7) solves inversion of the score ranking, which can be caused by an increment in the prediction score for an action class due to the change of the weight. It indicates that the increment in the score of a class can alter the rank of the scores in other classes. Namely, the extended subobjective function adjusts the error rate to be low and the fusion score summed over the entire class to be high. We also solve the minimization problem of the extended objective function Eq. (6) and the subobjective function Eq. (7) by employing SGD.[15] We refer to this approach as the extended class-wise and stream-wise fusion (ECSF).

4 Experimental Results

To evaluate the proposed method, we conduct extensive experiments on three action recognition datasets captured from websites and movies, namely UCF101,[11] HMDB51[12] toward accuracy, and Hollywood2[13] toward mean average precision (mAP).

4.1 *Experimental Settings*

4.1.1 *Implementation details*

All the parameters in ConvNets such as the size of minibatch (set to 256), momentum (set to 0.9), resizing ratio of input frames (set to the smaller side as 256), data augmentation method (e.g., random horizontal flipping), and the learning rate (initially set to 10^{-2}) are chosen to be the same as.[7] The initial weight $\mathbf{w}$ of the spatial stream for the ECSF is 0.3 over the entire class on the three datasets.

4.1.2 *Baselines*

We denote our proposed methods as class-wise and stream-wise fusion with the two-stream convolutional neural network (CSF + TSCNN) and extended CSF + TSCNN (ECSF + TSCNN). We also denote the CF with two-stream CNN (CF + TSCNN) and the stream-wise fusion with two-stream CNN (SF + TSCNN). We compare our ECSF method with three groups of baseline methods, the hand-crafted methods, the single stream CNN methods, and the two-stream CNN methods. Four hand-crafted methods include dense trajectory with multiview super vector (DT + MVSV) method,[16] improved dense trajectory with Fisher vector (iDT + FV) method,[2] the multiskip feature stacking (MIFS) which extracts features with multiple time skips obtained with multiple frequencies,[17] and spatial Fisher vector (SFV)[18] with MBH,[19] SIFT,[20] and MFCC.[21] The single stream CNN methods include the spatial stream CNN (SCNN) and the temporal stream CNN (TCNN). Four kinds of the two-stream CNN methods include early, late, and slow fusion with spatiotemporal CNN (STCNN) method,[6] combining VGG19 as the spatial stream and CNNM[7] as the temporal stream (VGG19 + CNNM) method,[8] the standard two-stream CNN (TSCNN) method,[7] and trajectory-pooled deep-convolutional descriptor (TDD).[9]

4.2 *Results*

4.2.1 *Stream-wise fusion*

We validate the performance of the stream-wise fusion (SF) method on UCF101, HMDB51, and Hollywood2 datasets. The results are shown in Fig. 2. The performance of the stream-wise fusion is significantly improved than that of conventional average fusion by balancing the spatial stream weight and the temporal stream weight. In addition, we can observe that the stream-wise fusion with 75%, 70%, and 65% of the temporal stream weight and 25%, 30%, and 35% of the spatial stream weight shows the best results on UCF101, HMDB51, and Hollywood2, respectively. The advanced performance by the stream-wise fusion in Fig. 2 confirms that a motion feature is more important than an appearance feature for the two-stream ConvNets on action recognition tasks.

4.2.2 *Class-wise fusion*

To evaluate the effectiveness of the proposed CF method, we qualitatively investigate the stability of our CF method on Hollywood2. As shown in Fig. 3, some classes such as *DriveCar*, *SitDown*, and *Run* have sharp variation in the

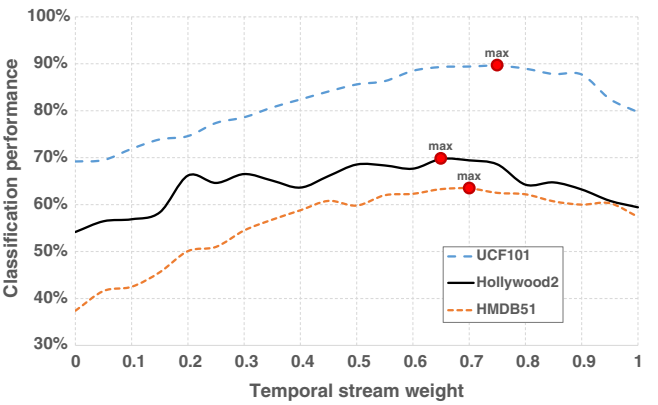

Fig. 2 Performance of stream-wise fusion on UCF101, HMDB51, and Hollywood2 datasets.

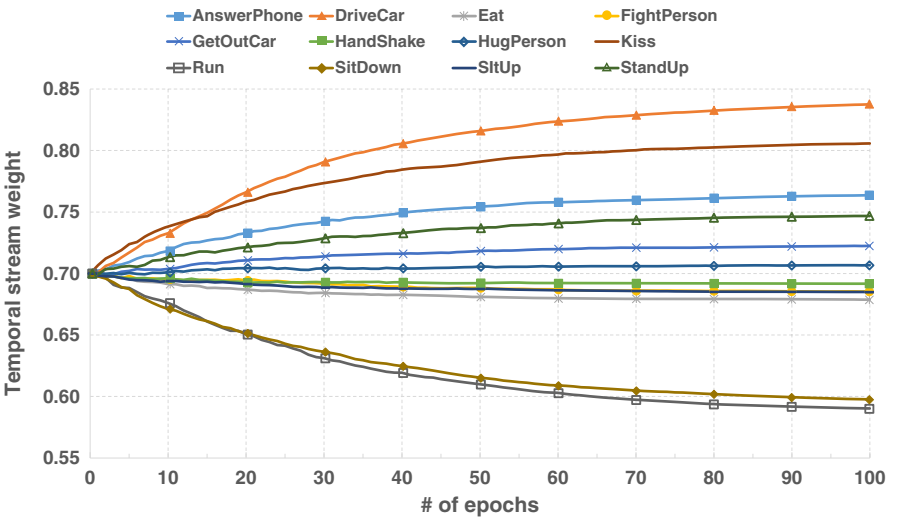

Fig. 3 Stability of the CF with regard to stream weights on Hollywood2. It is conducted with a temporal stream weight initially set to 0.7.

temporal stream weight at an early stage. However, the weight of each action class finally converges in 100 epochs (full pass through the training set). This means that the CF method can prevent a stream weight from fluctuating or diverging.

4.2.3 *Class-wise and stream-wise fusion*

We evaluate the performance of our proposed class-wise and stream-wise fusion (CSF) methods on UCF101, HMDB51, and Hollywood2 datasets. The results are summarized in Table 1. We implement two-stream ConvNets as our baseline (TSCNN*) based on the TSCNN,[7] then we combine the other proposed methods with TSCNN*. Regardless of the datasets, we see that our methods achieve the much better performance than TSCNN*. Specifically, the ECSF + TSCNN outperforms TSCNN* by 5.7%, 4.8%, and 3.6% on UCF101, HMDB51, and Hollywood2 datasets, respectively. In addition, we found an interesting fact that SF + TSCNN achieves better performance than CF + TSCNN on all the datasets. The reason is that SF + TSCNN is far less sensitive than CF + TSCNN to the initial weight of each stream. However, CF + TSCNN and SF + TSCNN are complementary and do not replace one another. It can be shown that CSF + TSCNN combined between CF + TSCNN and SF + TSCNN outperforms both CF + TSCNN and SF + TSCNN. In addition, we also evaluate our TSCNN with SVM fusion method. We can see that SVM + TSCNN achieves better performance by 0.6% for UCF101 and 2.0% for HMDB51 than TSCNN*. However, all of our proposed methods outperform SVM + TSCNN for all the three datasets by a large margin. We employ VGG16 network,[4] which has 16 layers (13 convolutional and 3 fully connected layers), instead of M2048,[7] which has 7 layers (5 convolutional and 2 fully connected layers). There could be a

Table 1 Performance of the class-wise and stream-wise fusion methods in accuracy (%) for UCF101 and HMDB51 and mAP (%) for Hollywood2.

Method	UCF101 (%)	HMDB51 (%)	Hollywood2 (%)
SCNN	69.2	37.4	54.2
TCNN	79.7	53.5	59.4
TSCNN*	85.6	59.8	68.5
SVM + TSCNN	86.2	61.8	68.4
CF + TSCNN	86.5	62.2	68.8
SF + TSCNN	89.6	63.5	69.7
CSF + TSCNN	89.9	64.2	71.9
ECSF + TSCNN	**91.3**	**64.6**	**72.1**

Table 2 Comparison of our proposed methods to the state-of-the-art in accuracy (%) for UCF101 and HMDB51 and mAP (%) for Hollywood2.

Method	UCF101 (%)	HMDB51 (%)	Hollywood2 (%)
SFV[18]	–	54.8	63.3
DT + MVSV[16]	83.5	55.9	–
iDT + FV[2]	85.9	57.2	64.3
MIFS[17]	89.1	**65.1**	**68.0**
SCNN	69.2	37.4	54.2
TCNN	79.7	53.5	59.4
STCNN[6]	65.4	–	–
VGG19 + CNNM[8]	87.7	–	–
TSCNN[7]	88.0	59.4	–
TDD[9]	**90.3**	63.2	–
TSCNN*	85.6	59.8	68.5
CSF + TSCNN	89.9	64.2	71.9
ECSF + TSCNN	**91.3**	**64.6**	**72.1**

performance gap between them due to the depth difference of each network; nonetheless our proposed fusion method is network-independent. Hence, it can be adopted to any other two-stream ConvNets. The improvement shown in Table 1 verifies that our class-wise and stream-wise fusion methods for two-stream ConvNets are more accurate than the existing average fusion method (TSCNN*).

4.2.4 *Comparison with the state-of-the-art*

Finally, we compare our recognition accuracy or mAP with the very recent state-of-the-art hand-crafted models[2,16–18] and deep-learned models.[6–9] Table 2 illustrates that our proposed CSF + TSCNN and ECSF + TSCNN methods are comparable to the state-of-the-art method[17] on HMDB51 and outperform the state-of-the-art[9,17] by a large margin (1.0% and 4.1%) on UCF101 and Hollywood2, respectively. Specifically, MIFS[17] which stacks features extracted with multiple time skips achieves a better result than ours on HMDB51 because MIFS covers a longer range of action sequences compared to TSCNN representations. This superior performance of ECSF + TSCNN indicates that the proposed method is highly valuable to two-stream ConvNets for enhancing a weighted fusion on action recognition tasks.

5 Conclusion

In this letter, we have proposed an ECSF method of two-stream convolutional networks for action recognition. Based on the supposition that appearance and motion information have different importance on each action, we cope with the unique properties of action classes by balancing the weight of a spatial stream and the weight of a temporal stream for each class. The extensive experiments demonstrate the superior performance of the proposed method on three datasets for action recognition. In the future, we will extend the proposed two-stream fusion method into the multistream fusion method, which handles more than two streams.

Acknowledgments

This work was supported by Institute for Information & Communications Technology Promotion (IITP) grant funded by the Korea government (MSIP) [No. B0101-16-0307, Basic Software Research in Human-level Lifelong Machine Learning (Machine Learning Center)].

References

1. I. Laptev, "On space-time interest points," *Int. J. Comput. Vision* **64**(2–3), 107–123 (2005).
2. H. Wang and C. Schmid, "Action recognition with improved trajectories," in *2013 IEEE Int. Conf. on Computer Vision (ICCV)*, IEEE, pp. 3551–3558 (2013).
3. A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, pp. 1097–1105 (2012).
4. K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint, arXiv:1409.1556* (2014).
5. S. Ji et al., "3D convolutional neural networks for human action recognition," *IEEE Trans. Pattern Anal. Mach. Intell.* **35**(1), 221–231 (2013).
6. A. Karpathy et al., "Large-scale video classification with convolutional neural networks," in *2014 IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, IEEE, pp. 1725–1732 (2014).
7. K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," in *Advances in Neural Information Processing Systems*, 568–576 (2014).
8. H. Ye et al., "Evaluating two-stream CNN for video classification," *arXiv preprint, arXiv:1504.01920* (2015).
9. L. Wang, Y. Qiao, and X. Tang, "Action recognition with trajectory-pooled deep-convolutional descriptors," *arXiv preprint, arXiv:1505.04868* (2015).
10. J. Deng et al., "Imagenet: a large-scale hierarchical image database," in *IEEE Conf. on Computer Vision and Pattern Recognition 2009 (CVPR 2009)*, IEEE, pp. 248–255, (2009).
11. K. Soomro, A. R. Zamir, and M. Shah, "Ucf101: a dataset of 101 human actions classes from videos in the wild," *arXiv preprint, arXiv:1212.0402* (2012).
12. H. Kuehne et al., "HMDB: a large video database for human motion recognition," in *2011 IEEE Int. Conf. on Computer Vision (ICCV)*, IEEE, pp. 2556–2563 (2011).
13. M. Marszalek, I. Laptev, and C. Schmid, "Actions in context," in *2009 IEEE Conf. on Computer Vision and Pattern Recognition (CVPR 2009)*, pp. 2929–2936, IEEE (2009).
14. R. Collobert and J. Weston, "A unified architecture for natural language processing: deep neural networks with multitask learning," in *Proc. of the 25th Int. Conf. on Machine Learning*, pp. 160–167, ACM (2008).
15. D. R. G. H. R. Williams and G. Hinton, "Learning representations by back-propagating errors," *Nature* **323**, 533–536 (1986).
16. Z. Cai et al., "Multi-view super vector for action recognition," in *2014 IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 596–603, IEEE (2014).
17. Z. Lan et al., "Beyond Gaussian pyramid: multi-skip feature stacking for action recognition," *arXiv preprint, arXiv:1411.6660* (2014).
18. D. Oneata, J. Verbeek, and C. Schmid, "Action and event recognition with fisher vectors on a compact feature set," in *2013 IEEE Int. Conf. on Computer Vision (ICCV)*, pp. 1817–1824, IEEE (2013).
19. N. Dalal, B. Triggs, and C. Schmid, "Human detection using oriented histograms of flow and appearance," in *European Conf. on Computer Vision (ECCV 2006)*, Springer, pp. 428–441 (2006).
20. D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vision* **60**(2), 91–110 (2004).
21. L. R. Rabiner and R. W. Schafer, "Introduction to digital speech processing," *Found. Trends Signal Process.* **1**(1), 1–194 (2007).