Postprint: Human Action Recognition Method Based on Improved Deep Convolutional Neural Network

Authors: Chen Shengdi, Wei Wei, He Bingqian, Chen Siyu, Liu Jiyuan

Date: 2018-05-02T00:00:00+00:00

Abstract

To address the problems of complex feature extraction and low recognition rates in existing action recognition algorithms, this paper proposes a network architecture that integrates batch normalization with the GoogLeNet network model. By applying the batch normalization concept from the image classification domain to the action recognition domain for training algorithm improvement, mini-batch normalization processing is implemented for network inputs of video action training samples. The method employs RGB images as input to the spatial network and optical flow as input to the temporal network, subsequently fusing the spatial-temporal networks to obtain the final action recognition results. Experiments on the UCF101 and HMDB51 datasets achieved accuracies of 93.50% and 68.32%, respectively. Experimental results demonstrate that the improved network architecture exhibits high recognition accuracy for video-based human action recognition.

Full Text

Human Action Recognition Method Based on Improved Deep Convolutional Neural Network

Chen Shengdi, Wei Wei, He Bingqian, Chen Siyu, Liu Jiyuan
(College of Computer Science & Technology, Chengdu University of Information Technology, Chengdu 610225, China)

Abstract: Existing action recognition algorithms suffer from complex feature extraction and low recognition accuracy. To address these issues, this paper proposes a network structure that combines batch normalization with the GoogLeNet model. By applying the batch normalization concept from image classification to action recognition, the training algorithm is improved through mini-batch normalization of network inputs for video action training samples.

The method uses RGB images as input to the spatial network and optical flow fields as input to the temporal network, then fuses the spatio-temporal networks to obtain final action recognition results. Experiments on the UCF101 and HMDB51 datasets achieve accuracies of 93.50% and 68.32%, respectively. The results demonstrate that the improved network architecture offers higher recognition accuracy for video-based human action recognition.

Keywords: action recognition; batch normalization; deep learning; convolutional neural network

0 Introduction

Human action recognition represents a significant research topic in computer vision with broad application value and research significance in areas such as video surveillance, content-based video retrieval, medical assistance, virtual reality, and intelligent human-computer interaction. Compared to static images, videos contain not only appearance information but also motion information. Consequently, action recognition performance is influenced by numerous factors, including varying lighting conditions in motion scenes, camera viewpoints, background clutter, and differences in action poses.

Current action recognition methods can be broadly categorized into two classes: (a) traditional action recognition methods, and (b) convolutional neural network-based action recognition methods. Traditional approaches primarily analyze RGB image sequences. For instance, Shen et al. [5] proposed combining depth information with RGB images for behavior recognition. Zhang et al. [6] utilized histograms of spatio-temporal gradients and optical flow for feature description combined with K-means clustering. Shotton et al. [7] employed Harris and Gabor detectors to identify spatio-temporal interest points, constructing 3D histograms of oriented gradients (HOG3D) for feature representation and proposing a human action recognition method based on colored spatio-temporal interest points. Ofli et al. [8] introduced the Sequence of Most Informative Joints (SMIJ) for feature representation. Chen et al. [9] captured motion information using Depth Motion Maps (DMMs) from front, side, and top views, then applied Local Binary Patterns (LBP) for feature representation. Zhao et al. [10] proposed a feature extraction method combining dense optical flow trajectories with a sparse coding framework (DOF-SC) for action recognition. Li et al. [11] developed a feature learning algorithm based on a single-layer regularized optical flow-constrained autoencoder.

Convolutional neural network-based action recognition methods focus on constructing more effective network architectures. Simonyan et al. [12] proposed a two-stream network structure, demonstrating that convolutional neural networks trained with inter-frame optical flow features can achieve excellent performance even with limited datasets. He et al. [13] utilized spatial pyramid pooling by adding a pooling layer after the final convolutional layer to pool output

features, enabling convolutional neural networks to accept inputs of non-fixed sizes. Wang et al. [14] constructed a network with 3D convolutional kernels and max pooling for automatic RGB-D video recognition. Wang et al. [15] adjusted several mainstream network structures, proposing very deep two-stream convolutional neural networks for video action recognition. Wang et al. [16] combined convolutional neural networks with SVM support vector machines to recognize five daily human actions captured by intelligent terminals. Han [17] proposed a two-modality action recognition method, using convolutional neural networks for static information captured by Kinect sensors and recurrent neural networks for dynamic information, finally fusing features from both models for action recognition.

Shallow learning networks have limited capacity to represent complex functions and exhibit significant generalization limitations when training samples are scarce. Simonyan et al. [18] validated on large-scale datasets that recognition performance improves substantially when convolutional neural network depth increases to 16-19 weight layers. The GoogLeNet network [19] incorporates multiple inception module structures based on traditional deep convolutional neural networks [20]. This paper proposes a network architecture combining batch normalization with the GoogLeNet model for video-based human action recognition, improving upon traditional deep convolutional neural networks in both training algorithm and network structure. The spatial stream network captures appearance information from video frame RGB images, while the temporal stream captures motion information through optical flow fields between consecutive frames. Fusing spatio-temporal networks considers both appearance and motion information to improve action recognition accuracy. This paper also investigates the impact of different dropout rates in Dropout layers and different linear weighted fusion ratios between spatial and temporal networks on recognition accuracy.

1 Network Architecture Improvement

Deep neural networks face internal covariate shift during training, where the input distribution of each layer is affected by parameters from previous layers. As network depth increases, minor changes in layer parameters are amplified, potentially causing gradient vanishing or explosion. As network parameters are continuously updated, input ranges across layers vary and change, slowing convergence and potentially leading to suboptimal local optima. These issues arise from internal covariate shift [21]. To mitigate these side effects, one can modify network structure, apply whitening processing in activation layers, or change parameter optimization algorithms [22-25]. To address these problems, this paper adapts the batch normalization algorithm proposed in [26] for ImageNet image classification to the action recognition domain, applying mini-batch normalization to video action training sample inputs.

1.1 Batch Normalization Algorithm

Traditional batch normalization is shown in Equation (1):

$$\hat{x} = \frac{x - \mathbb{E}[X]}{\sqrt{\text{var}[X]}}$$

where x represents the input vector of a network layer and X represents the entire training set input collection. Equation (1) shows that batch normalization output depends on the input and the entire training sample values. In the training set, each layer' s input x is generated from the previous layer' s output, influenced by model parameters. Therefore, when updating network parameters via backpropagation, the Jacobian matrix for γ and β corresponding to batch normalization must be computed, as shown in Equation (2).

Applying batch normalization to every layer' s input would be computationally expensive (requiring covariance matrix calculation). To address this, Ioffe et al. [26] proposed two simplified improvements:

a) Simplified improvement: Perform independent batch normalization on each input dimension rather than joint normalization, as shown in Equation (3):

$$\hat{x}^{(k)} = \frac{x^{(k)} - \mathbb{E}[x^{(k)}]}{\sqrt{\text{var}[x^{(k)}]}}$$

where $x^{(k)}$ represents the k-th dimension of the input sample; $\mathbb{E}[x^{(k)}]$ and $\text{var}[x^{(k)}]$ represent the mean and variance of the input, respectively. Lécun et al. [27] demonstrated that even when training features are uncorrelated, the batch normalization algorithm in Equation (3) can accelerate convergence, but it may alter the original representation of each layer, preventing the input from completely expressing the original output features. Therefore, to ensure the introduced batch normalization transformation remains an identity function, a pair of parameters $\gamma^{(k)}, \beta^{(k)}$ must be added to each input $x^{(k)}$, as shown in Equation (4):

$$y^{(k)} = \gamma^{(k)}\hat{x}^{(k)} + \beta^{(k)}$$

where $\sqrt{\text{var}[x^{(k)}]}$ represents the standard deviation of the input, equivalent to scaling transformation of input $x^{(k)}$, while $\beta^{(k)}$ is equivalent to translation transformation of $x^{(k)}$. These two parameters, like neural network model parameters, are learned through training to restore model representation capability.

b) Simplified improvement: In stochastic gradient training, mini-batch samples are used for training. Mean and variance are computed for each layer on each mini-batch, enabling neural network statistics (variance and

mean) calculated during batch normalization to be used in gradient back-propagation. Assuming mini-batch size is m and a certain dimension of layer input is $B = \{x_{1...m}\}$, dimension-wise normalization is performed as shown in Equation (5):

$$\hat{x}_i = \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}}$$

Algorithm 1 describes the normalization transformation algorithm inserted into a layer of a deep convolutional neural network, where the network input changes from x to $BN_{\gamma,\beta}(x)$. Algorithm 2 shows the algorithm flow for inserting batch normalization transformation into the entire deep convolutional neural network. Algorithm 1 demonstrates that normalization processing includes normalizing inputs and performing scale-invariant translation transformation on normalized data.

Algorithm 1: Mini-Batch Normalization Transform Algorithm
Input: Mini-batch input values $B = \{x_{1...m}\}$; parameters to be learned γ, β
Output: Normalized and transformed values $y_i = BN_{\gamma,\beta}(x_i)$

1. Compute mini-batch mean: $\mu_B \leftarrow \frac{1}{m}\sum_{i=1}^{m} x_i$
2. Compute mini-batch variance: $\sigma_B^2 \leftarrow \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu_B)^2$
3. Normalize: $\hat{x}_i \leftarrow \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}}$
4. Scale and shift: $y_i \leftarrow \gamma\hat{x}_i + \beta$

Algorithm 2: Neural Network Batch Normalization Transform Algorithm
Input: Neural network N, training parameter set θ; input collection for each layer $\{x^{(k)}\}_{k=1...K}$
Output: Batch-normalized network

1. Initialize $BN_N \leftarrow N$
2. For each layer $k = 1...K$ do:
   - Replace original input $x^{(k)}$ with affine-transformed input $y^{(k)} = BN_{\gamma^{(k)},\beta^{(k)}}(x^{(k)})$
3. End for
4. Train BN_N to update network parameters: $\theta \cup \{\gamma^{(k)}, \beta^{(k)}\}_{k=1...K}$
5. Freeze parameters and output batch-normalized network

1.2 Batch Normalization Combined with GoogLeNet Network Construction

This paper proposes a network structure combining batch normalization with the GoogLeNet model for video-based human action recognition. The specific processing applies batch normalization to input features of each convolutional layer, then feeds the normalized features into ReLU activation layers. Figure 1 [Figure 1: see original paper] shows the inception network structure with batch normalization applied to an inception layer in GoogLeNet.

In addition to adding batch normalization layers after each convolutional layer in inception modules as shown in Figure 1, the improved network model also includes a batch normalization layer after every convolutional layer in the bottom network, followed by a ReLU activation layer before subsequent network layers. The deep convolutional neural network structure applied to human action recognition is shown in Table 1 .

2 Two-Stream Network

2.1 Two-Stream Network Architecture

Videos consist of spatial and temporal components. The spatial component contains scene and object appearance information in individual frames, while the temporal component includes camera and object motion information. The spatio-temporal two-stream network model is shown in Figure 2 [Figure 2: see original paper].

The spatial stream network input is RGB images, while the temporal stream input is optical flow fields. The deep convolutional neural network with added normalization requires backpropagation to compute loss function gradients and parameters introduced by batch normalization transformation. Equation (6) presents the gradient computation process for network parameters using backpropagation.

2.2 Network Training

Public action recognition datasets are relatively small compared to ImageNet [28]. When convolutional neural networks are deep, small training sets can lead to overfitting. Therefore, preprocessing is performed using data augmentation techniques to expand the training set, and network pre-training is used for weight initialization.

Data augmentation involves geometric transformations of the training dataset to increase its size. Since random cropping tends to select central image regions, causing overfitting, this paper crops corner and center regions of image frames to enhance scale diversity. Specifically, input data is fixed to size 256×340, then a candidate crop size is randomly selected from {256, 224, 192} for cropping, and the cropped region is resized to 224×224.

Pre-training: The deep convolutional neural network is pre-trained on the ImageNet dataset. Since the spatial network input is RGB images, it can be directly pre-trained on ImageNet. For the temporal network input of 10 stacked optical flow fields, network adjustments are required. This paper uses TV-L1 optical flow [29], first extracting optical flow fields from action videos using OpenCV, then discretizing optical flow to the [0, 255] interval to match RGB ranges, and finally averaging the first-layer filters of the ImageNet pre-trained spatial network model across channels and replicating the result 20 times (for vertical and horizontal optical flow directions) to initialize the temporal network.

Network Training: Due to ImageNet pre-training, a smaller learning rate is used during training. The momentum value is set to 0.9. For the spatial network, the base learning rate is 0.001, reduced to 1/10 every 1,800 iterations, with a maximum of 5,000 iterations. The temporal network learning rate is 0.003, reduced to 1/10 at 15,000 iterations, reduced again at 18,000 iterations, and training completes at 20,000 iterations.

3 Experiments

3.1 Experimental Data

The experimental data uses public video action recognition datasets UCF101 [30] and HMDB51 [31]. Sample action illustrations are shown in Figure 3 [Figure 3: see original paper].

The UCF101 dataset contains 101 action categories with 13,320 video clips. UCF101 consists of web videos captured in unconstrained real-world environments with low frame resolution, containing varied lighting information, partial occlusions, and camera motion. The dataset categorizes actions into five types: (a) human-human interaction (5 categories: haircut, head massage, etc.); (b) playing instruments (10 categories: playing flute, violin, etc.); (c) body motion only (16 categories: blowing candles, Tai Chi, etc.); (d) human-object interaction (20 categories: hair drying, chopping, etc.); (e) sports (50 categories: billiards, breaststroke, etc.).

The HMDB51 dataset contains 51 action categories with 6,849 video clips, mostly from movie clips and partially from YouTube. HMDB51 is similarly divided into five types: (a) facial expressions with object interaction (3 categories: smoking, drinking, etc.); (b) general facial actions (4 categories: smiling, talking, etc.); (c) body motion with human interaction (7 categories: hugging, kissing, etc.); (d) body motion with object interaction (18 categories: sword drawing, horse riding, etc.); (e) general body motion (19 categories: clapping, handstand, etc.).

3.2 Experimental Results and Analysis

Experiments were conducted on a Caffe platform with a single GPU under Linux. Following the dataset split standards specified in [30,31], three train/test splits

were used, with each split dividing the dataset into approximately 70% training and 30% testing sets.

Dropout layers are commonly added to deep convolutional neural networks to prevent overfitting [32]. This paper investigates the impact of different dropout_ratio parameter values on recognition accuracy in the newly constructed spatio-temporal action recognition network. Experiments were conducted on UCF101 split1 with various dropout rates, with results shown in Table 2 .

Table 2 shows action recognition accuracy on UCF101 split1 for different dropout_ratio values. For the temporal network, a dropout rate of 0.7 achieves 0.22% and 0.3% higher accuracy than rates of 0.4 and 0.6, respectively. For the spatial network, a dropout rate of 0.8 achieves 1.07% and 0.52% higher accuracy than rates of 0.4 and 0.6, respectively. Subsequent experiments use dropout rates of 0.7 and 0.8 for temporal and spatial networks, respectively.

Figure 4 [Figure 4: see original paper] shows the training iteration convergence curves for spatio-temporal networks on UCF101 split3. For the spatial stream (Figure 4(a)), accuracy approaches 80% at 1,000 iterations with rapidly decreasing training loss, after which the accuracy curve gradually rises and the loss curve gradually declines. After 2,000 iterations, accuracy remains above 80% with loss below 0.5, showing stable convergence. For the temporal stream (Figure 4(b)), accuracy approaches 80% at 1,500 iterations with rapidly decreasing training loss, after which the accuracy curve gradually rises and the loss curve gradually declines. After 15,000 iterations, accuracy remains above 80% with training loss below 0.3, showing stable convergence.

Table 3 shows the recognition accuracy of the improved spatio-temporal network on UCF101 and HMDB51 datasets across three splits. Motion information extracted by the temporal stream network achieves higher recognition rates than appearance information from the spatial stream, demonstrating that motion information is more important than appearance information for action recognition tasks.

Table 4 shows the recognition accuracy of the fused spatio-temporal network. Linear weighted fusion of temporal and spatial network classification results yields final recognition rates. With classification confidence weights set to 1:1, the fused two-stream convolutional neural network outperforms weight ratios of 1:1.2 and 1:1.5. Compared to Table 3, the fused spatio-temporal two-stream deep convolutional neural network effectively improves accuracy over individual networks.

Table 5 compares the proposed method with typical action recognition methods on UCF101 and HMDB51. Improved dense trajectories [4,33] and IDT with higher-dimensional encoding [34] use traditional hand-crafted features. Two-stream [15] constructs a two-stream network but with shallow architecture. Very deep two-stream [18] improves network depth. KVMF [35] uses multiple 3D volumes as network inputs. The proposed improved fused spatio-temporal

two-stream deep convolutional neural network achieves better action recognition performance on these datasets, obtaining 93.50% and 68.32% accuracy on UCF101 and HMDB51, respectively.

4 Conclusion

This paper proposes an improved deep convolutional neural network model structure for human action recognition tasks and constructs a spatio-temporal two-stream deep convolutional neural network architecture. Fine-tuned on the ImageNet dataset, the fused deep convolutional neural network achieves 93.50% and 68.32% recognition rates on UCF101 and HMDB51 datasets, respectively. While deep convolutional neural network algorithms have been successfully applied in pattern recognition research, they remain distant from real-time commercial applications primarily due to lengthy training times. Future work should focus on parallel computing algorithms for deep convolutional neural networks.

References

[1] Jhuang H, Serre T, Wolf L, et al. A biologically inspired system for action recognition [C]// Proc of IEEE International Conference on Computer Vision. 2007: 1-8.
[2] Karpathy A, Toderici G, Shetty S, et al. Large-scale video classification with convolutional neural networks [C]// Proc of Computer Vision and Pattern Recognition. 2014: 1725-1732.
[3] Ling Peipei, Qiu Song, Cai Mingming, et al. Human action recognition with privileged information [J]. Journal of Image and Graphics, 2017, 22 (4): 482-491.
[4] Wang H, Schmid C. Action recognition with improved trajectories [C]// Proc of IEEE International Conference on Computer Vision. 2013: 3551-3558.
[5] Shen Xiaoxia, Zhang Hua, Gao Zan, et al. Behavior recognition algorithm based on depth information and RGB images [J]. Pattern Recognition and Artificial Intelligence, 2013, 26 (8): 722-218.
[6] Zhang Jie, Wu Jianzhang, Tang Jiali, et al. Human action recognition method based on spatio-temporal image segmentation and interactive region detection [J]. Application Research of Computers, 2017, 34 (1): 302-305, 320.
[7] Shotton J, Fitzgibbon A, Cook M, et al. Real-time human pose recognition in parts from single depth images [C]// Proc of Computer Vision and Pattern Recognition. 2011: 1297-1304.
[8] Ofli F, Chaudhry R, Kurillo G, et a1. Sequence of the most informative joints (SMIJ): a new representation for human skeletal action recognition [J]. Journal of Visual Communication and Image Representation, 2014, 25 (1): 24-38.
[9] Chen C, Jafari R, Kehtarnavaz N. Action recognition from depth sequences using depth motion maps-based local binary patterns [C]// Proc of Applications of Computer Vision. 2015: 1092-1099.
[10] Zhao Xiaojian, Zeng Xiaoqin. Behavior recognition method based on

dense optical flow trajectories and sparse coding algorithm [J]. Computer Applications, 2016, 36 (1): 181-187.
[11] Li Yawei, Jin Lizuo, Sun Changyin, et al. Action recognition based on optical flow constrained autoencoder [J]. Journal of Southeast University: Natural Science Edition, 2017, 47 (4): 691-696.
[12] Simonyan K, Zisserman A. Two-stream convolutional networks for action recognition in videos [J]. Advances in Neural Information Processing Systems, 2014, 1 (4): 568-576.
[13] He K, Zhang X, Ren S, et al. Spatial pyramid pooling in deep convolutional networks for visual recognition [J]. IEEE Trans on Pattern Analysis and Machine Intelligence, 2015, 37 (9): 1904-1916.
[14] Wang K, Wang X, Lin L, et al. 3D human activity recognition with reconfigurable convolutional neural networks [C]// Proc of ACM International Conference on Multimedia. 2015: 97-106.
[15] Wang L, Xiong Y, Wang Z, et al. Towards good practices for very deep two-stream ConvNets [J]. arXiv preprint arXiv: 1507. 02159, 2015.
[16] Wang Zhongmin, Cao Hongjiang, Fan Lin. A human behavior recognition method based on convolutional neural network deep learning [J]. Computer Science, 2016, 43 (s2): 56-58.
[17] Han Minjie. Multi-modal action recognition based on deep learning framework [J]. Computer and Modernization, 2017 (7): 48-52.
[18] Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition [J]. arXiv preprint arXiv: 1409. 1556, 2014.
[19] Szegedy C, Liu W, Jia Y, et al. Going deeper with convolutions [C]// Proc of Computer Vision and Pattern Recognition. 2015.
[20] Lecun Y, Boser B, Denker J S, et al. Backpropagation applied to handwritten zip code recognition [J]. Neural Computation, 1989, 1 (4): 541-551.
[21] Shimodaira H. Improving predictive inference under covariate shift by weighting the log-likelihood function [J]. Journal of Statistical Planning and Inference, 2000, 90 (2): 227-244.
[22] Wiesler S, Richard A, Schluter R, et al. Mean-normalized stochastic gradient for large-scale deep learning [C]// Proc of IEEE International Conference on Acoustics, Speech and Signal Processing. 2014: 180-184.
[23] Raiko T, Valpola H, Lecun Y. Deep learning made easier by linear transformations in perceptrons [C]// Proc of the 15th Internation Conference on Artificial Intelligence and Statistics. 2012, 22: 924-932.
[24] Povey D, Zhang X, Khudanpur S. Parallel training of deep neural networks with natural gradient and parameter averaging [J]. arXiv preprint arXiv: 1410. 7455, 2014.
[25] Desjardins G, Simonyan K, Pascanu R, et al. Natural neural networks [J]. Computer Science, 2015, 22 (8): 847-856.
[26] Ioffe S, Szegedy C. Batch normalization: accelerating deep network training by reducing internal covariate shift [C]// Proc of the 32nd International Conference on Machine Learning. 2015: 448-456.
[27] Lécun Y, Bottou L, Bengio Y, et al. Gradient-based learning applied to document recognition [J]. Proceedings of the IEEE, 1998, 86 (11): 2278-2324.

[28] Deng J, Dong W, Socher R, et al. ImageNet: a large-scale hierarchical image database [C]// Proc of Computer Vision and Pattern Recognition. 2009: 248-255.
[29] Pérez J S. TV-L1 optical flow estimation [J]. Image Processing on Line, 2013, 2 (4): 137-150.
[30] Soomro K, Zamir A R, Shah M. UCF101: a dataset of 101 human actions classes from videos in the wild [J]. arXiv preprint arXiv: 1212. 0402, 2012.
[31] Kuehne H, Jhuang H, Stiefelhagen R, et al. HMDB51: a large video database for human motion recognition [C]// Proc of IEEE International Conference on Computer Vision. 2012: 2556-2563.
[32] Hinton G E, Srivastava N, Krizhevsky A, et al. Improving neural networks by preventing co-adaptation of feature detectors [J]. Computer Science, 2012, 3 (4): 212-223.
[33] Wang H, Schmid C. LEAR-INRIA submission for the thumos workshop [C]// Proc of ICCV Workshop on THUMOS Challenge. 2013.
[34] Peng X, Wang L, Wang X, et al. Bag of visual words and fusion methods for action recognition: Comprehensive study and good practice [J]. Computer Vision and Image Understanding, 2016, 150 (C): 109-125.
[35] Zhu W, Hu J, Sun G, et al. A key volume mining deep framework for action recognition [C]// Proc of Computer Vision and Pattern Recognition. 2016: 1991-1999.

Note: Figure translations are in progress. See original paper for figures.