A Key Volume Mining Deep Framework for Action Recognition

Wangjiang Zhu[1], Jie Hu[2], Gang Sun[2], Xudong Cao[2], Yu Qiao[3]

[1] Tsinghua University [2] SenseTime Group Limited
[3] Shenzhen Institutes of Advanced Technology, CAS, China

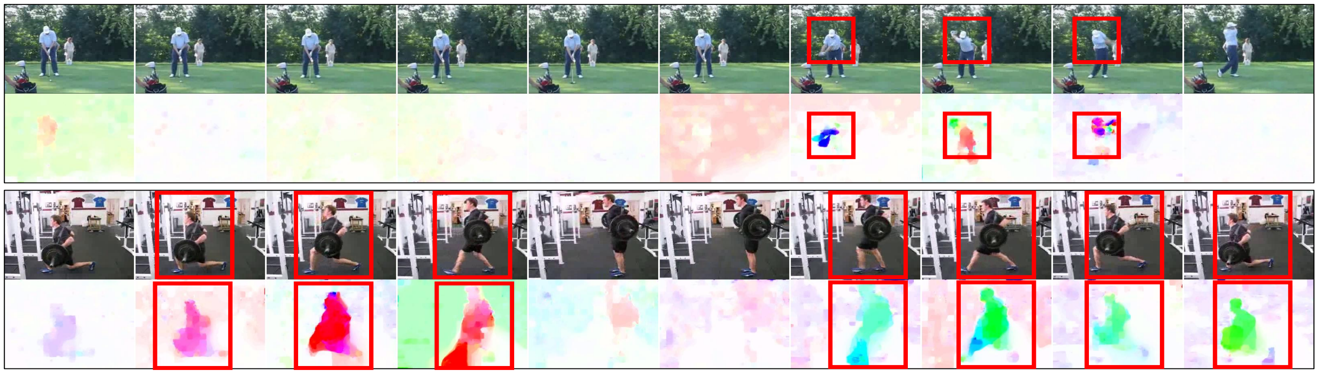

Figure 1. Key volumes detected by our key volume mining deep framework. A volume is a spatial-temporal video clip. The top row shows key volumes are very sparse among the whole video, and the second row shows that key volumes may come from different modalities (different motion patterns here). Note that frames are sampled with fixed time interval.

Abstract. *Recently, deep learning approaches have demonstrated remarkable progresses for action recognition in videos. Most existing deep frameworks equally treat every volume i.e. spatial-temporal video clip, and directly assign a video label to all volumes sampled from it. However, within a video, discriminative actions may occur sparsely in a few key volumes, and most other volumes are irrelevant to the labeled action category. Training with a large proportion of irrelevant volumes will hurt performance.*

To address this issue, we propose a key volume mining deep framework to identify key volumes and conduct classification simultaneously. Specifically, our framework is trained is optimized in an alternative way integrated to the forward and backward stages of Stochastic Gradient Descent (SGD). In the forward pass, our network mines key volumes for each action class. In the backward pass, it updates network parameters with the help of these mined key volumes. In addition, we propose "Stochastic out" to model key volumes from multi-modalities, and an effective yet simple "unsupervised key volume proposal" method for high quality volume sampling. Our experiments show that action recognition performance can be significantly improved by mining key volumes, and we achieve state-of-the-art performance on HMDB51 and UCF101 (93.1%).

1. Introduction

Action recognition in videos receives extensive research interests nowadays due to its wide applications in video retrieval, surveillance, human-computer interface [21] etc. Early works [16, 29, 30] utilized hand-crafted spatial-temporal local descriptors for video representation and classification. Inspired by the remarkable successes of deep learning for image classification, recent works [23, 13, 28] have explored deep convolutional neural networks (CNN) for video classification.

A problem exists when extending deep learning methods from image to video: unlike images, videos are 3D in nature and have variable temporal durations, but CNN only accept fixed size input. Existing works [23, 13, 28] tackle this problem by sampling fixed-size volumes regardless of the video's actual length. Note that a volume is a spatial-temporal video clip containing a sequence of images cropped from consecutive frames.

However, it is problematic to equally treat all the sampled volumes and apply video-level labels to all of them. Because of limited memory and computational resources, a volume can only cover very limited pixels (e.g. 224x224x10) comparing to a long video. Such small volumes are more likely to be irrelevant to or less relevant to the action categories at video level. As shown in Figure 1, a video generally contains one or several key volumes which are discriminative for action recognition. This phenomenon is also observed in [22, 3, 2]. Assigning video label to all sampled volumes as in [23, 13, 28] will bring in large proportion of noises and hurt the final performance.

We argue that action recognition in videos is actually a weakly supervised learning problem as only video level labeling is available, and it is necessary to find out key volumes for better classification. In this work, we propose a unified deep learning framework for simultaneously identifying discriminative key volumes and training classifiers free from the harm of irrelevant volumes. The two objectives are optimized alternatively through EM-like loops integrated in SGD training. Specifically, in the forward pass, we feed a bag of volumes into our network, then mine key volumes for each action class based on the response matrix; in the backward pass, we update network parameters with the help of those key volumes.

In addition, we propose two novel techniques to further improve our deep framework. First, we propose *Stochastic out* to select key volumes from multiple modalities; Second, we design an effective yet simple unsupervised key volume proposal algorithm to improve the probability that an input bag contains key volumes. Experimental results show that our deep framework and the proposed techniques significantly improve action recognition performance.

The main contributions of this paper can be summarized as follows: 1) We propose an end-to-end deep framework to simultaneously identify key volumes and do action classification. And we integrate the alternative optimization into forward and backward stages of SGD training. 2) We propose two novel techniques, i.e., *Stochastic out* and *unsupervised key volume proposal* to benefit the deep framework. 3) With the proposed deep framework and novel techniques, we finally achieve excellent performance (93.1%) on the well-known UCF101 [25] benchmark.

2. Related Work

Deep learning based action recognition. Alex's notable work [14] starts the booming of deep learning in computer vision community. Since then, many computer vision areas are rapidly evolving. For image classification, recent works [27, 24, 7, 9] have demonstrated going deep is critical to achieve high performance, and [7, 9] have surpassed human-level performance in the challenging ImageNet [4] classification task.

These successes inspire researchers to exploit deep neural networks for video classification [11, 13, 23, 28, 17, 33]. An early work [11] extended convolutional neural network to 3D for action recognition, but only examined the proposed models on small datasets. Similarly, Tran et al. [28] trained 3-Dimensional Convolutional Networks (C3D) on a large dataset and achieved state-of-the-art performance. To explicitly model motion pattern, Simonyan and Zisserman [23] proposed the two-stream architecture which consists of RGB and optical flow streams to capture the appearance and motion information respectively. Following this pipeline, Wang et al. [33] showed deeper networks can benefit action recognition. Because of deep learning's appetite for huge amount of data, Karpathy et al. developed the One Million sports video dataset [13] using weak tag label from Youtube.

However, unlike image tasks, deep learning did not yield significant improvement on videos over traditional methods such as the notable iDT descriptor + Fisher vector approach [29]. We argue that there are two reasons to account for this fact. First, most public action datasets such as UCF101 and HMDB51 have much smaller scales than ImageNet, both in term of the numbers of samples and categories. Second, actions are weakly labeled at video level because of prohibitive cost for detail spatial-temporal annotations. Existing works [13, 23, 28, 33] suffer from the weak supervision issue as they directly assign video labels to very small volumes. In this paper, we tackle this issue by learning key volumes and doing classification simultaneously.

More recently, Recurrent Neural Networks are explored to model the temporal structure of videos [17, 26, 34]. This paper distinguishes from these works and only focus on volume-level classification. And we believe the improvement of volume level classifier will also benefit sequential models as they are built upon volume level CNN features.

Multiple instance learning. Our work shares similar spirit with multiple instance learning (MIL) [1]. According to MIL theory, one training sample is a bag of instances. A positive bag contains at least one positive instance, while a negative bag contains only negative instances. For object detection, recent works [19, 20, 8] adopted MIL framework to mine discriminative objects (part) for each class, and thus do classification and localization at the same time. For action recognition, [22] and [3] applied mi-SVM [2] to find discriminative 3D cuboids. Despite of sharing similar motivations with our work, they are based on shallow models and are completely different from our deep framework.

3. Our Approach

In this section, we will first illustrate our motivations and then present the proposed deep framework together with Stochastic out operation, finally we will show the unsupervised key volume proposal method.

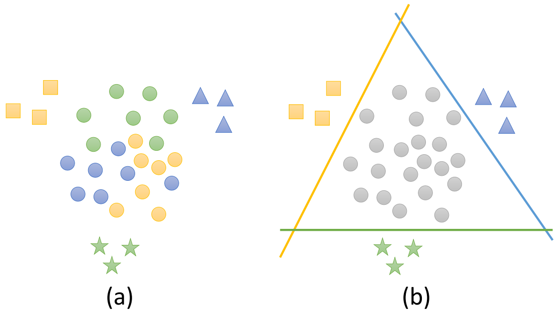

Figure 2. A toy example: the large number of irrelevant samples (circles) will prevent us to learn classifiers which capture the essential characteristics (triangles, squares and stars) for each class (denoted by 3 different colors). If we can label all irrelevant samples, we will get better one-vs-all classifiers which treating all circles as negative samples.

3.1. Motivation

As observed in the Introduction, the key volumes which are discriminative for action recognition are relatively rare compared to irrelevant volumes. In this case, it is problematic for existing deep learning approaches [23, 13, 28] to directly assign video labels to volumes. As illustrated in Figure 2(a), the large number of irrelevant samples (circles) will prevent us to learn classifiers which capture the essential characteristics for the three classes.

Ideally, if all irrelevant volumes could be identified in advance, we can avoid the ill-posed target and achieve a well-formulated classification task. As shown in Figure 2(b), we have N *one-vs-all* classifiers, each of them only fires at its corresponding key volumes. Without noisy irrelevant volumes, we can achieve much better results in this ideal case.

Practically, the ideal case is rare due to two obstacles: the prohibitive cost and the inherent ambiguity to manually annotate key volumes for actions. In the following, we will present our deep learning approach to simultaneously identify key volumes and train volume-level classifiers with weak video-level category label.

3.2. Key Volume Mining Deep Framework

To mine key volumes, we need a good volume classifier; And to train a good volume classifier, we need key volumes. This is a chicken-and-egg problem, as we do not have well-labeled key volumes in advance. We solve this problem using an alternative optimization method, and integrate the processes to the forward and backward steps of SGD. Specifically, we identify key volumes for each class in the forward pass and update network parameters using key volumes in the backward pass.

Following multiple instance learning convention [1], our training samples are bags of volumes (instances). The hypothesis underlying key volume mining is that a bag contains at least one key volume. Suppose key volume ratio for a video is r, then the probability for a K-sized bag to contain at least one key volume is $1-(1-r)^K$. As K increases, this probability grows toward 1. Given a classifier of moderate quality, key volumes tend to have higher response scores, and thus we can do key volume mining based on those scores. Updating network parameters with these selected key volumes, we can achieve classifiers focusing on the discriminative volumes for each action class.

Inspired by the toy example in Figure 2, we learn N volume-level binary classifiers, where N is the number of categories. Each of them only responses to the key volumes of one specific category, while rejecting key volumes of other categories and all irrelevant volumes. As shown in Figure 3, our deep neural network receives a bag of 3D volumes as input. Those volumes are convolved through the shared CNN module, and then passed to N logistic regressors (sigmoid) to get a score matrix S. Formally, we represent the score matrix as:

$$S = \{S_{k,n}\}, \quad k = 1, ..., K, \quad n = 1, ..., N, \qquad (1)$$

where k is the volume index, K is the total number of volumes; n is the class label, N is the total number of classes. $S_{k,n}$ denotes the response of the n-th binary classifier at the k-th volume.

Based upon this response matrix, we mine key volumes for each action category. Given a bag with label Y, for classifiers $n \neq Y$, all volumes are expected to response low, so we can minimize the max response in the bag; for classifier $n = Y$, we hope the key volumes have high responses. This loss can be formulated as

$$l = -\sum_n ([n = Y]\text{log}p_n + [n \neq Y]\text{log}(1 - p_n)), \qquad (2)$$

where p_n is a function of $S_{:,n}$, *i.e.* responses of all volumes at classifier n., Y is the video label, and $[.]$ is an indicator function.

We define p_n as:

$$p_n = \begin{cases} \text{MaxOut}(S_{:,n}), & n \neq Y, \\ \text{StochasticOut}(S_{:,n}), & n = Y. \end{cases}$$

Max out is defined in [6]. It outputs the max value within the input vector and thus is a deterministic operator. In order to avoid converging to a dominant key volume modality, we propose *Stochastic Out*.

Stochastic out is the counterpart of max out, and is different from Stochastic Pooling [36] that conducting pooling for a local image patch. It randomly chooses a number from a vector with a probability proportional to the value of this number. Suppose the input vector is x, the probability of

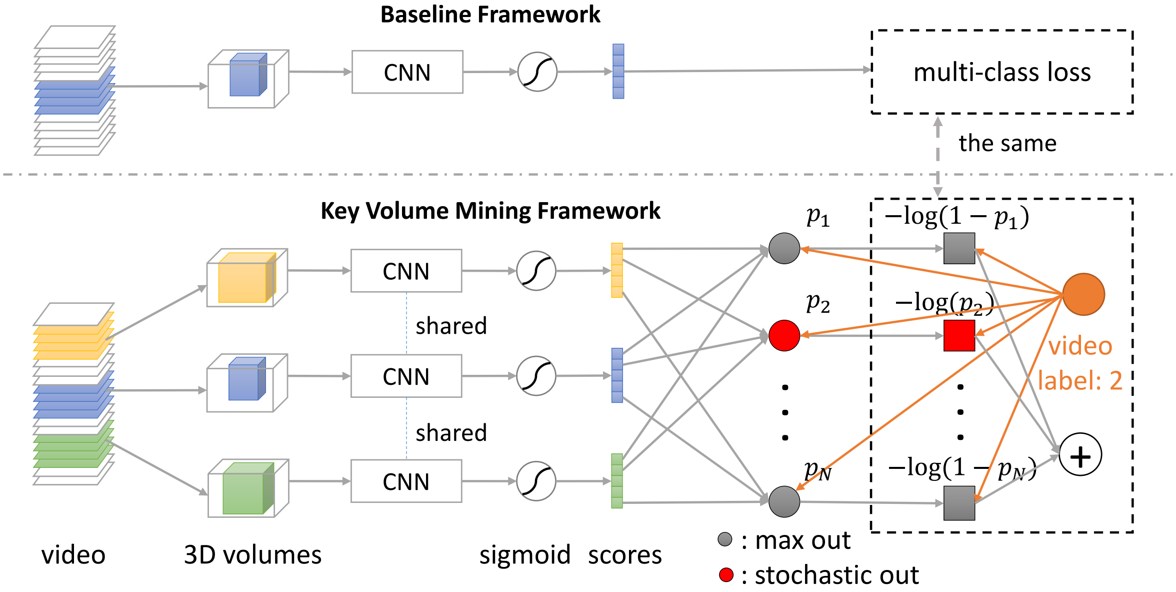

Figure 3. The proposed key volume mining deep framework (bottom) and the baseline framework (top). Unlike the baseline framework which directly assign video labels to volumes, in the proposed framework, we simultaneously mine key volumes and do action classification. The network is optimized alternatively. For each SGD mini-batch, in the forward pass, we mine key volumes according to their response scores; and in the backward pass, we update network parameters with the help of mined key volumes.

choosing x_i via stochastic out is equal to

$$x_i / \sum x_j \tag{3}$$

Note that all elements in x are non-negative. This requirement is guaranteed by sigmoid function in our framework.

By stochastic out at $n = Y$, we can select key volumes for the n-th category and then update the corresponding one-vs-all classifier.

An alternative method for selecting key volumes is max out, *i.e.* selecting the volume with the maximum response. It works well if all key volumes come from a single modality. However, it is inferior if key volumes come from multiple modalities. Max out may only select the largest modality while suppress the rest smaller ones. In other words, the selected volumes of max out are biased and incomplete.

Stochastic out addresses the shortcoming of max out. By introducing randomness, stochastic out selects higher response volumes with higher probabilities. This mechanism allows us to select the rest smaller modalities and reject irrelevant or noisy volumes with very low responses.

The empirical comparison between stochastic out and max out will be shown in Table 1.

Discussion on training strategy. As aforementioned, we solve the chicken-and-egg problem in EM-like loops. A good starting point is critical to make sure the EM-like loop converge to a satisfied result. We empirically found that naively running the iterative loop from scratch fails to yield good performance. In this work, we first assign video-level labels to all volumes, and train the baseline convolutional network in Figure 3 with large learning rate until the loss stops to decrease; then we use this 1 stage pre-trained network to initialize the CNN module in our deep framework and start running the iterative loop. It is worth noting that we do not do further pre-training with smaller learning rate. This is because, after too many stages of pre-training, the CNN network will overfit to training data, and all the volume responses are very high thus are less indicative for key volume mining. In Table 2, we show the proposed training strategy is superior to directly train from scratch or pre-train the baseline model with too many stages.

3.3. Unsupervised Key Volume Proposal

In training, we feed bags of volumes as inputs. To get training volumes, a straight forward baseline is random sampling a 3D cuboid of fixed spatial-temporal size from the video.

As analysis in Section 3.2, the hypothesis underlying key volume mining is that a bag contains at least one key volume with high probability. To get higher probability $1-(1-r)^K$, we can either use larger K or improve key volume ratio r. K is constraint by GPU memory and computational capacity, and could not be too large. In our experiments, typically $K = 6$. Thus improving r is meaningful for the success of key volume mining.

With this in mind, we study the common characteristics

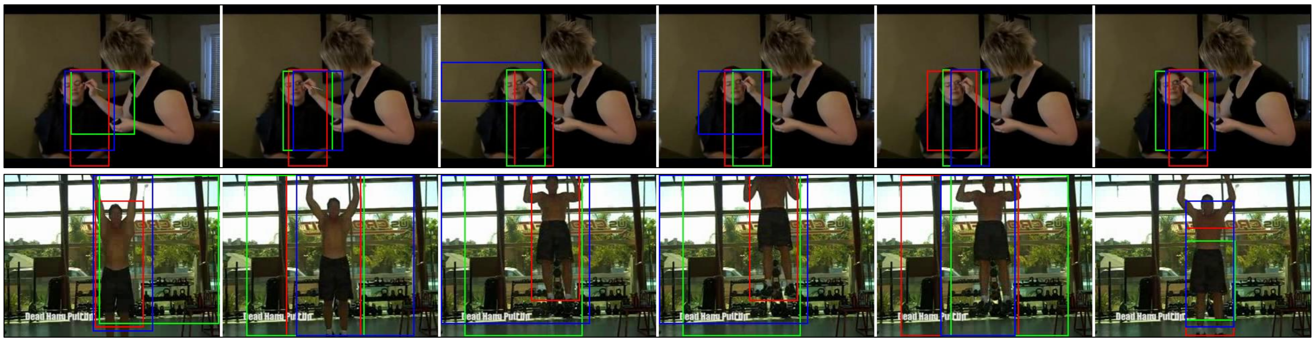

Figure 4. Unsupervised key volumes proposal visualization. We show top-3 scored bounding boxes every 10 frames. In the first line, the proposals capture the moving part (hand) of the actor; In the second line, the proposals tend to capture the whole actor

of selected key volumes, and find that the selected key volumes are highly correlated with the motion of actors.

In this work, we simply extend the edge box method [37] to 3D video as its score function encourages high intensity of motion boundary. It may yield better results by utilizing more sophisticated methods [5, 18], but exploring the best proposal methods for action recognition is out of the scope of this paper. In this work, our main purpose is to show that good unsupervised key volumes proposal will benefit the proposed deep framework.

We represent a proposal as a tuple (frame-id, box-id), which indicates a 3D volume starting at frame-id with spatial location specified by box-id. As mentioned above, each volume extend T consecutive frames in temporal dimension, where T is the fixed temporal size. The proposal algorithm is described below:

1. Generate a bounding box set which covers various sizes and aspect ratios in a sliding window fashion. This set is shared by all frames in the same video.

2. Apply edge-box scoring function to optical flow images for all bounding boxes in all frames.

3. Average scores for each bounding box id using a 1D sliding window along the temporal dimension. The length of the sliding window is set to be $T = 10$.

4. Do temporal non-maximum suppression for each bounding box id. Finally, we end up with a pool of candidate proposals.

At training stage, for each mini-batch, we randomly choose K candidates from the pool by importance sampling, i.e., the probability of sampling a candidate is proportional to its score.

In Figure 4, we visualize the top-3 scored bounding boxes of two actions at sampled frames. As we can see, the proposals mainly focus on moving regions as they exhibit strong motion boundaries. Note that a key volume proposal could be a part of an actor, an object in interaction or the whole moving person.

3.4. Implementation Details

We use both RGB data and optical flow data for action recognition in a two-steam [23, 33, 17] fashion. For the motion stream, We use off-the-shell tvl1 method [35] implemented in OpenCV to compute optical flows.

We use a modified parallel Caffe [12] to train our deep neural network, and specifically, 4 Titan GPUs are used for the parallel computing. We use SGD to optimize our neural network and each mini-batch contains 64 videos (bags), 288 volumes ($K = 6$). We use initial learning rate 0.001 for the RGB stream, and larger learning rate 0.005 for the flow stream. We train the proposed deep framework for three stages, iterating 12000, 8000, and 5000 times respectively, and shrink the learning rate by a factor of $1/10$ when moving to next stage.

Frames are first cropped using our unsupervised key volume proposals. The cropped volumes are then resized to a fixed size (e.g. 224x224). For rgb stream, we use single frame volume ($T_{\text{rgb}} = 1$), and we stack 10 consecutive frames for flow stream ($T_{\text{flow}} = 10$) just the same as [23]. And because each flow field is a 2 channel image, the number channel of flow volume is $T_{\text{flow}} \times 2 = 20$. For RGB stream, we directly fine-tune from a pre-trained ImageNet model, and for motion stream, we fine-tune from a revised pre-train model using channel repeating for the first convolution kernel as illustrated in [33].

Following the same settings of previous works [23, 28, 17]. In testing, prediction scores of the two streams are weighted averaged (1/3 for RGB stream, 2/3 for motion stream) to generate a volume-level prediction. Then we averagely aggregate predictions of 250 (equal to the number of volumes when doing 10 views testing at 25 temporal locations [23]) sampled volumes to get the video-level prediction.

Network	Volume sampling strategy	Bag size K	Accuracy	
			rgb	flow
1) baseline CNN	random spat. temp.	1	82.1	85.4
2) key-volume mining CNN	random spat.	6	84.0	86.6
3) key-volume mining CNN	random temp.	6	82.5	87.6
4) key-volume mining CNN	random spat. temp.	6	84.1	87.9
5) key-volume mining CNN	unsupervised proposal	1	82.2	86.4
6) key-volume mining CNN	unsupervised proposal	3	84.0	88.6
7) key-volume mining CNN	unsupervised proposal	6	84.8	89.0
8) key-volume mining CNN	unsupervised proposal	12	85.3	89.2
9) key-volume mining CNN, all max-out	unsupervised proposal	6	84.5	88.5

Table 1. Ablation studies conducted on UCF101 split1. “Baseline CNN” and “key-volume mining CNN” correspond to the top row and bottom row of Figure 3 respectively. We use GoogLeNet with batch normalization [10] as CNN modules shown in Figure 3.

4. Experiments

In this section, we first conduct ablation experiments to validate the proposed framework and novel techniques, and then compare with recent state-of-the-art works.

Datasets. We conduct experiments on UCF101 [25] and HMDB51 [15]. UCF101 is one of the biggest action datasets with 13,320 videos distributed in 101 classes. It is widely evaluated in previous works [30, 23, 13, 28, 17]. HMDB51 is a very challenging dataset which contains 6,766 videos distributed in 51 classes. Both datasets give 3 train/test splits, and we follow these splits in our experiments.

4.1. Key Volume Mining Benefit Classification

In this subsection, we verify the effectiveness of learning key volumes. First we re-implement two-stream [23, 33] works whose network structure is shown at top of Figure 3. This baseline randomly samples spatial-temporal volumes and directly transfers video label to those sampled volumes. Our key volume mining network is built upon this baseline, but uses bag of volumes as input, and adds max/stochastic out operations for key volume mining. For a fair comparison, both our framework and the baseline framework use the same batch size and train at the same iterations. As shown in experiment 1) and 4) in Table 1, key volume mining significantly improves performances for both rgb and flow streams (2.0% and 2.5% respectively).

4.2. Random Volume Sampling Comparison

In this subsection, we compare different volume sampling strategies for bag composition. As mentioned previously, a volume is a 3D video clip, and can be denoted as a tuple (frame-id, bbox). We compare three different volume sampling strategies: (1) random spatial sampling with frame-id fixed; (2) random temporal sampling with spatial bounding box fixed; (3) random spatial-temporal sampling in the joint space. Comparing experiment 2)-3) with 1) in Table 1, we find that flow stream benefits from both random spatial sampling and random temporal sampling. In contrast, rgb stream favors only random spatial sampling. This is because rgb images look similar within a short video, random temporal sampling will lead to close-identical volumes for a bag, which making it hard to ensure at least one key volume per-bag. Experiment 4) in Table 1 shows that random spatial-temporal sampling yields better performance than constraint sampling strategies.

4.3. Unsupervised Key Volume Proposal

In section 3.3, we designed an unsupervised key volume proposal method to generate bags which are more likely to contain key volumes. Compare experiment 7) with 4) in Table 1, we find unsupervised key volume proposal is obviously better than random sampling for both rgb and flow streams.

4.4. The influence of bag size

The underlying hypothesis of our method is a bag contains at least one key volume. As analyzed in previous section, bag size directly influences the probability of including key volumes. In Table 1, we compare classification accuracies at different bag size (K). As expected, the performance increases as K increases. To balance the computational cost, we fix $K = 6$ for experiments in other sections.

Note that when $K = 1$, our network degenerates to the baseline network in Figure 3. The only difference between experiment 5) and 1) lies in volume sampling strategy (importance sampling using proposal score vs. random spatial-temporal sampling). This comparison shows, even without key volume mining, unsupervised key volume proposal still helps CNN based action classification. This is because the proposal score itself is a weak indicator of key volume, and we are more likely to sample key volumes in training.

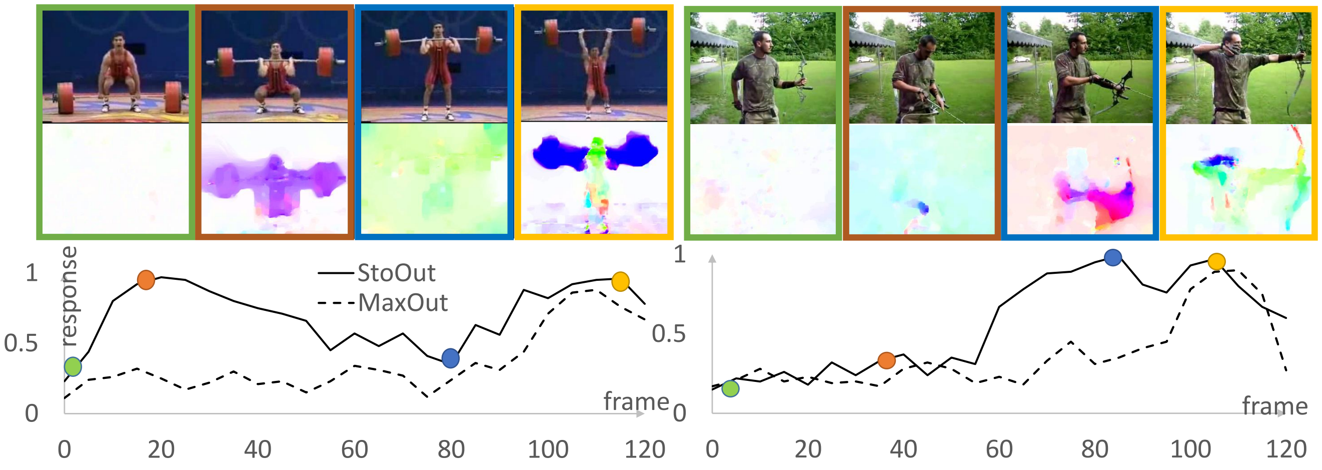

Figure 5. Compare response curves between stochastic out and max out on training data. The first row shows sampled volumes of two actions. The second row shows the response curves. The color bounding rectangle in the first row and the color dots in the second row shows the temporal correspondence. We can see that stochastic out can learn key volumes from multiple modalities, while max out only learns one type of key volumes.

4.5. Stochastic Out vs. Max Out

As analysis in Sec 3.2, stochastic out has the ability to mine key volumes from multiple modalities. Herein, we experimentally validate this analysis under two settings: using stochastic out or using max out for identifying key volumes. As experiment 9) in Table 1 shows, max out is inferior to stochastic out for key volume mining.

To visualize the difference between stochastic out and max out, we do test-on-training and show the response curve. Figure 5 compares the response curves of the two strategies on the same video. To integrate out spatial variance, we average 5 view testing scores as frame score, and then draw the response curve along temporal dimension. As we can see, stochastic out responses highly at key volumes of various modalities while max out only responses highly at one dominant key volume type.

4.6. Network Initialization Comparison

As we train the proposed deep framework in an EM-like loop, a good initialization is crucial. In this subsection, we compare three different initialization strategies: (1). random initialization; (2). pre-train the baseline network with large learning rate for one stage, and use it for initialization; (3). the same as (2), but pre-training for three stages until the baseline model converges.

random init.	pre-train 1 stage	pre-train 3 stages
85.0	89.0	88.5

Table 2. The influence of various initialization strategies. (On flow stream of split 1, UCF101)

As shown in Table 2, the best result is obtained by one stage pre-training. As discussed before, we believe that random initialization is not enough to guarantee good convergence, while over pre-training prone to over-fitting, both key or irrelevant volumes have strong response, trapping our deep framework into a sub-optimal local minimal.

4.7. Action Localization via Key Volume Classifier

With the key volumes mining network, we can do rough action localization. In Figure 6, we show the response heat maps generated using volume responses. Specifically, first, we randomly sample 200 volumes from a video, then score those volumes using the learnt key volume mining model, and finally we define each pixel's response value as average of responses of volumes which contain this pixel.

4.8. Comparison with State-of-the-art

We compare two stream performances averaged on all 3 splits with recent state-of-the-art methods, especially deep learning based methods, on both UCF101 and HMDB51 datasets. We use GoogLeNet with batch normalization [10] as our CNN module.

On UCF101, we significantly surpass the recent deep learning approaches, including very expensive VGG16 models [33].

On HMDB51 [15], our result is on par with the previous best [32]. We conjecture this is because the training data size of HMDB51 is much smaller than that of UCF101 [25], such small data size does not favor end-to-end deep approaches.

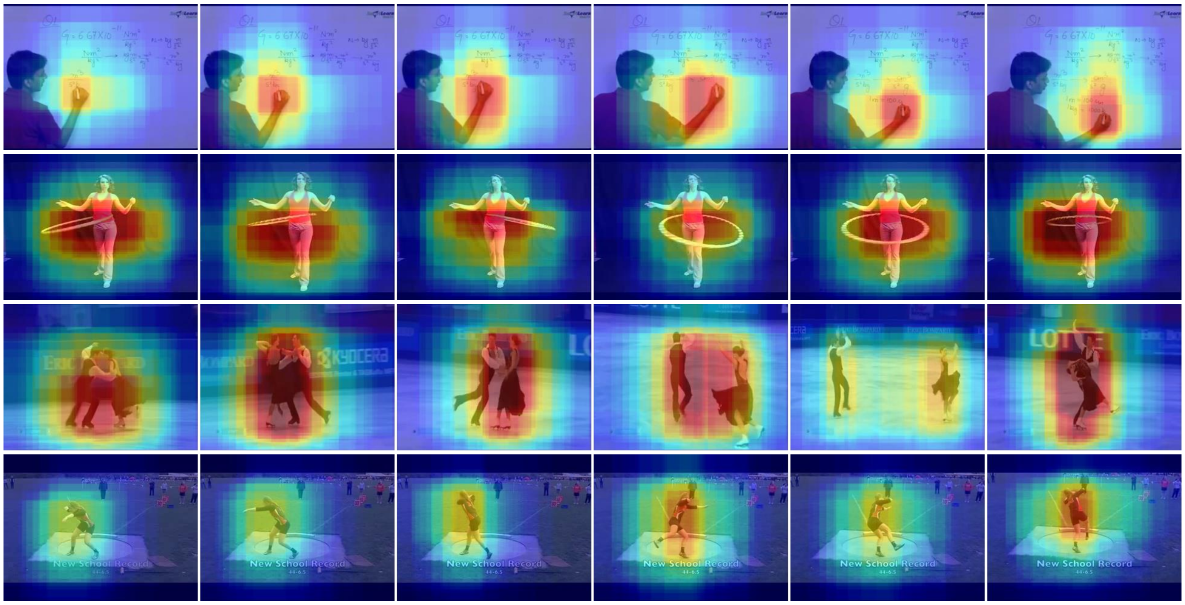

Figure 6. Action heat maps generated from scores of randomly sampled volumes. As we can see, the top two rows show our model has successfully localized the writing hand and hula-loop; the third row shows our model localize two person dancing together as a key volume to this class, and gives lower score when the two persons separate; the last row shows that our model focused on some distinct moments within a continuous action.

Method	Accuracy (%)
iDT [31]	85.9
C3D [28]	82.3
Ng *et al.*[17]	88.6
TDD [32]	90.3
Two stream [23]	86.9
Wang *et al.*, use GoogleNet [33]	89.3
Wang *et al.*, use VGG16 [33]	91.4
Ours	**93.1**

Table 3. Comparison with state-of-the-art methods on UCF101 [25], three splits averaged.

Method	Accuracy (%)
iDT [30]	57.2
TDD [32]	63.2
Two stream [23]	58.0
Ours	**63.3**

Table 4. Comparison with state-of-the-art methods on HMDB51 [15], three splits averaged.

5. Conclusions

In this paper, we proposed a deep framework to simultaneously mine discriminative key volumes and do action classification. Experiments showed this deep framework achieves better classification performance than previous CNN based methods.

We also proposed stochastic out to handle key volumes from multi-modalities. Both experimental results and response curve visualization proved stochastic out can deal with multi-modality key volumes.

In order to feed higher quality bags to the deep framework, we proposed an effective yet simple unsupervised volume proposal method. Experiments showed it significantly improves performance.

References

[1] J. Amores. Multiple instance classification: Review, taxonomy and comparative study. *Artificial Intelligence*, 201:81–105, 2013. 2, 3

[2] S. Andrews, I. Tsochantaridis, and T. Hofmann. Support vector machines for multiple-instance learning. In *Advances in Neural Information Processing Systems 15*, pages 561–568. MIT Press, 2003. 2

[3] B. Antic, T. Milbich, and B. Ommer. Less is more: Video trimming for action recognition. In *The IEEE International Conference on Computer Vision (ICCV) Workshops*, December 2013. 2

[4] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. IEEE, 2009. 2

[5] K. Fragkiadaki, P. Arbelaez, P. Felsen, and J. Malik. Learning to segment moving objects in videos. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015. 5

[6] I. J. Goodfellow, D. Warde-farley, M. Mirza, A. Courville, and Y. Bengio. Maxout networks. In *In ICML*, 2013. 3

[7] K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. *arXiv preprint arXiv:1502.01852*, 2015. 2

[8] L. Hou, D. Samaras, T. M. Kurç, Y. Gao, J. E. Davis, and J. H. Saltz. Efficient multiple instance convolutional neural networks for gigapixel resolution image classification. *CoRR*, abs/1504.07947, 2015. 2

[9] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015. 2

[10] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *CoRR*, abs/1502.03167, 2015. 6, 7

[11] S. Ji, W. Xu, M. Yang, and K. Yu. 3d convolutional neural networks for human action recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(1):221–231, 2013. 2

[12] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014. 5

[13] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale video classification with convolutional neural networks. In *CVPR*, 2014. 1, 2, 3, 6

[14] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012. 2

[15] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre. HMDB: a large video database for human motion recognition. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2011. 6, 7, 8

[16] I. Laptev. On space-time interest points. *International Journal of Computer Vision*, 64(2-3):107–123, 2005. 1

[17] J. Y. Ng, M. J. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and G. Toderici. Beyond short snippets: Deep networks for video classification. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pages 4694–4702, 2015. 2, 5, 6, 8

[18] D. Oneata, J. Revaud, J. Verbeek, and C. Schmid. Spatio-temporal object detection proposals. In *ECCV 2014 - European Conference on Computer Vision*, Zurich, Switzerland, Sept. 2014. Springer. 5

[19] M. Oquab, L. Bottou, I. Laptev, and J. Sivic. Is object localization for free? - weakly-supervised learning with convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015. 2

[20] G. Papandreou, I. Kokkinos, and P.-A. Savalle. Modeling local and global deformations in deep learning: Epitomic convolution, multiple instance learning, and sliding window detection. June 2015. 2

[21] R. Poppe. A survey on vision-based human action recognition. *Image and vision computing*, 28(6):976–990, 2010. 1

[22] M. Sapienza, F. Cuzzolin, and P. H. Torr. Learning discriminative space-time action parts from weakly labelled videos. *International Journal of Computer Vision*, 2014. 2

[23] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 568–576. Curran Associates, Inc., 2014. 1, 2, 3, 5, 6, 8

[24] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 2

[25] K. Soomro, A. R. Zamir, and M. Shah. UCF101: A Dataset of 101 Human Action Classes From Videos in The Wild. Technical report, Nov. 2012. 2, 6, 7, 8

[26] N. Srivastava, E. Mansimov, and R. Salakhutdinov. Unsupervised learning of video representations using lstms. *arXiv preprint arXiv:1502.04681*, 2015. 2

[27] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. *arXiv preprint arXiv:1409.4842*, 2014. 2

[28] D. Tran, L. D. Bourdev, R. Fergus, L. Torresani, and M. Paluri. C3D: generic features for video analysis. *CoRR*, abs/1412.0767, 2014. 1, 2, 3, 5, 6, 8

[29] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu. Dense trajectories and motion boundary descriptors for action recognition. *International journal of computer vision*, 103(1):60–79, 2013. 1, 2

[30] H. Wang and C. Schmid. Action recognition with improved trajectories. In *IEEE International Conference on Computer Vision*, Sydney, Australia, 2013. 1, 6, 8

[31] H. Wang and C. Schmid. Lear-inria submission for the thumos workshop. *ICCV Workshop on Action Recognition with a Large Number of Classes*, 2013. 8

[32] L. Wang, Y. Qiao, and X. Tang. Action recognition with trajectory-pooled deep-convolutional descriptors. In *CVPR*, pages 4305–4314, 2015. 7, 8

[33] L. Wang, Y. Xiong, Z. Wang, and Y. Qiao. Towards Good Practices for Very Deep Two-Stream ConvNets. *ArXiv e-prints*, July 2015. 2, 5, 6, 7, 8

[34] Z. Wu, X. Wang, Y.-G. Jiang, H. Ye, and X. Xue. Modeling spatial-temporal clues in a hybrid deep learning framework for video classification. *Proceedings of ACM Multimedia*, 2015. 2

[35] C. Zach, T. Pock, and H. Bischof. A duality based approach for realtime tv-l1 optical flow. In *In Ann. Symp. German Association Patt. Recogn*, pages 214–223, 2007. 5

[36] M. D. Zeiler and R. Fergus. Stochastic pooling for regularization of deep convolutional neural networks. *CoRR*, abs/1301.3557, 2013. 3

[37] C. L. Zitnick and P. Dollár. Edge boxes: Locating object proposals from edges. In *ECCV*, 2014. 5