Temporal Hallucinating for Action Recognition with Few Still Images

Yali Wang[1]* Lei Zhou[1,3]* Yu Qiao[1,2]†
[1] Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, China
[2] The Chinese University of Hong Kong [3] SenseTime Group Limited

Abstract

Action recognition in still images has been recently promoted by deep learning. However, the success of these deep models heavily depends on huge amount of training images for various action categories, which may not be available in practice. Alternatively, humans can classify new action categories after seeing few images, since we may not only compare appearance similarities between images on hand, but also attempt to recall importance motion cues from relevant action videos in our memory. To mimic this capacity, we propose a novel Hybrid Video Memory (HVM) machine, which can hallucinate temporal features of still images from video memory, in order to boost action recognition with few still images. First, we design a temporal memory module consisting of temporal hallucinating and predicting. Temporal hallucinating can generate temporal features of still images in an unsupervised manner. Hence, it can be flexibly used in realistic scenarios, where image and video categories may not be consistent. Temporal predicting can effectively infer action categories for query image, by integrating temporal features of training images and videos within a domain-adaptation manner. Second, we design a spatial memory module for spatial predicting. As spatial and temporal features are complementary to represent different actions, we apply spatial-temporal prediction fusion to further boost performance. Finally, we design a video selection module to select strongly-relevant videos as memory. In this case, we can balance the number of images and videos to reduce prediction bias as well as preserve computation efficiency. To show the effectiveness, we conduct extensive experiments on three challenging data sets, where our HVM outperforms a number of recent approaches by temporal hallucinating from video memory.

1. Introduction

Action recognition in still images has been an active research topic in computer vision, due to its wide applications in image retrieval, human-computer interaction, and so on [10]. Recent advances in this task are mainly driven by deep learning models [2, 7, 8, 18, 41], based on the remarkable successes of convolutional neural networks (CNNs) in image classification [12, 17, 29, 32, 38]. However, the power of those deep models is built upon huge amounts of training images for various action categories, which may not be always available for realistic applications.

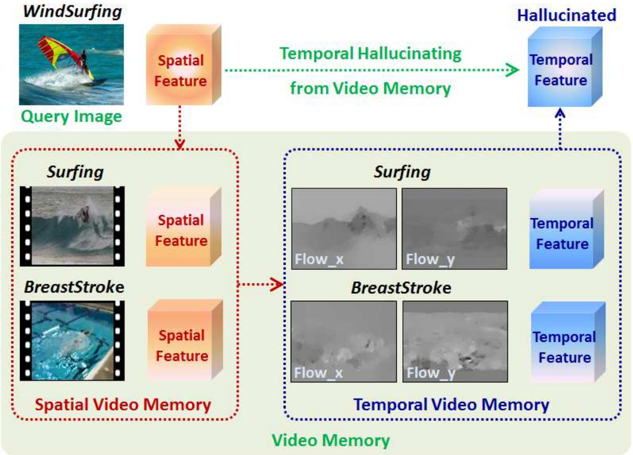

Figure 1. Temporal hallucinating (better view in color). First, we compare similarities between spatial features of query image and spatial video memory. Based on this, we hallucinate temporal features of query image from temporal video memory.

Alternatively, humans can correctly understand the new action concept in a query image, after checking out a few images. Our key insight is that, in the few-image scenario, humans may not only compare appearance similarity between still images on hand, but also attempt to recall temporal motions of relevant actions from memory. For example, when we see an image about windsurfing in Fig. 1, we intuitively refresh the surfing motions of relevant actions which we have ever seen from olympic videos, TV and so on. Motivated by this, we propose to address action recognition with few images, via hallucinating motion cues of still images from videos. Recently, several deep learning models have been proposed to predict optical flow (i.e., an important motion cue) by RGB video frames [3, 23, 26, 42].

*Equally-contributed first authors ({yl.wang, lei.zhou}@siat.ac.cn).
†Corresponding author (yu.qiao@siat.ac.cn).

However, these approaches may be infeasible for our problems. First, most of these approaches require at least two RGB frames of a video as inputs to generate the corresponding optical flow. This is unapplicable for a single still image. Second, our goal is to classify complex actions in the wild. Hence, instead of predicting low-level optical flow, it may be preferable to infer high-level representations for action recognition, especially when few training images are available. Finally, these approaches only focus on the video domain itself, which may not be effective to handle domain difference between image and video in our problem.

To address the challenges above, we propose a novel Hybrid Video Memory (HVM) machine in this paper, which can hallucinate temporal features of still images from relevant videos in memory, in order to boost action recognition with few training images. **First**, we design a temporal memory module consisting of temporal hallucinating and predicting. Temporal hallucinating can learn high-level temporal features of still images from video memory, via comparing spatial similarities between images and videos. Since hallucination is generated in an unsupervised manner, it can be widely applied to realistic scenarios, where action categories between images and videos may be different. Next, temporal predicting is to infer action categories of query images, according to temporal features. Since our design flexibly takes domain difference between images and videos into account, we can leverage temporal features of both training images and video memory together to boost temporal prediction. **Second**, we design a spatial memory module for spatial predicting. As spatial and temporal features are complementary to represent actions, we fuse spatial and temporal prediction to further enhance the performance of HVM. **Finally**, we design a video selection module to select strongly-relevant videos as memory. In this case, we can balance the number of videos and training images to reduce prediction bias as well as preserve computation efficiency. To show the effectiveness, we conduct extensive experiments on three challenging data sets. Our results show that, HVM outperforms a number of recent works with a higher classification accuracy, by temporal hallucinating of still images from video memory.

2. Related Works

Action Recognition. The recent advances of action recognition in visual data are mainly driven by deep learning [2, 4, 5, 8, 10, 15, 28, 33]. Compared to video-based action recognition, image-based action recognition is a more challenging task, due to large appearance variations and lack of motion descriptions in the wild images [10]. Most existing approaches mainly take advantage of available spatial information such as scene-object contexts [8, 41], or human parts-poses-attributes [2, 7, 18], to recognize actions in still images. However, the absence of action motions may restrict these deep models, especially when the training set is scarce (i.e., only little spatial information is available). Alternatively, video-based action recognition has been intensively investigated in the recent years [15, 22, 28, 31, 33, 36],with the rise of large-scale video benchmarks [13, 15, 30].Besides of spatial modality, videos contain a temporal modality (i.e., optical flow), which provides discriminative motion cues to improve action recognition [28]. Recently, several deep models have been proposed to generate optical flows from RGB video frames [3, 20, 23, 26, 35, 42],and applied to action recognition in videos [23, 42]. However, these approaches may be infeasible for still images. First, these models require at least two RGB frames of a video as inputs, which do not exist for a single still image. In addition, raw optical flows learned from these video-based approaches may not be effective for images, due to domain difference between image and video.

Deep Learning with Limited Data. The remarkable success of deep learning heavily relies on the availability of large-scale visual benchmarks. Hence, its power is often limited, when the target data set is scarce in many real problems. On the contrary, human can learn new concepts from very little supervision [11, 19, 25, 39]. Inspired by this fact, a number of few-shot learning approaches have been proposed by Bayesian program learning [19], siamese neural networks [1, 16], memory machines [9, 14, 27, 34], and so on. Especially, the recent memory machines achieve promising performance with few training images, by mimicking the learning procedure of humans with visual attention mechanisms. However, the complex network structure in these approaches may lead to training instability. More importantly, query and memory in most memory machines are from the same domain, which can lead to a biased learning procedure when domain shifts exist (like our image-video case). Alternatively, a well-known approach to handle domain difference is transfer learning, where one can use limited target data to fine-tune deep models pretrained on large-scale source data [40]. However, these general transfer learning approaches may ignore the important knowledge of specific tasks (such as temporal action cues in our case), and thus reduce the performance in our problem.

3. Hybrid Video Memory (HVM) Machine

Humans can correctly learn the action concept in a query image, after checking out few training images. Our key insight is that, when quite a few images are available, humans may not only compare appearance similarity between still images on hand, but also attempt to recall temporal motions of relevant actions from memory. To mimic this capacity, we introduce a novel hybrid video memory (HVM) machine. Even though few training images are available, our HVM Net can effectively hallucinate their temporal motions from video memory (e.g., the existing video benchmarks

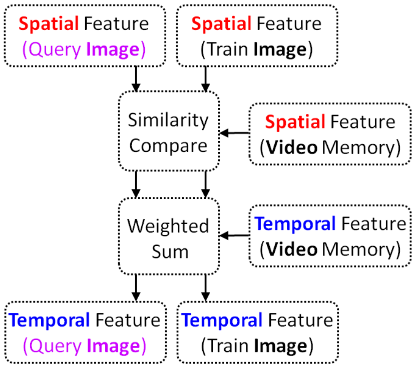

Figure 2. Temporal hallucinating in temporal memory module (better view in color). First, we compare spatial similarities between images and videos. Then, we hallucinate temporal features of images unsupervisedly, via weighted sum over temporal features of videos. More explanations can be found in Section 3.2.1.

such as UCF101), and subsequently boost action recognition via spatial-temporal integration.

3.1. Action Representation in Videos and Images

To represent distinct action characteristics in videos and images, we choose deep neural network as our feature generator in HVM. Specifically, we train a popular two-steam CNN architecture in action recognition [36, 37], by feeding RGB and optical flow of video memory respectively into spatial and temporal streams. As a result, we obtain spatial and temporal features of videos,

$$\{\mathbf{V}^{rgb},\ \mathbf{V}^{flow}\}. \tag{1}$$

Next, we feed the query and training images into spatial CNN, and obtain their spatial features,

$$\mathbf{U}^{rgb}_{all} = \{\mathbf{u}^{rgb}_{*},\ \mathbf{U}^{rgb}\}, \tag{2}$$

where $\mathbf{u}^{rgb}_{*}$ refers to the query image, and $\mathbf{U}^{rgb}$ refers to the training images (quite a few in our case). Note that, still images do not have optical flows originally, and thus no temporal features are available for them. In the following, we propose a novel temporal memory module, which can hallucinate temporal features of still images, and integrate them within video memory for temporal prediction.

3.2. Temporal Memory Module

In fact, spatial features may be limited to classify highly-confused actions, especially when we only have a glance at static human gestures in such a few images (e.g., one image per category). Alternatively, temporal features often contain important motion cues of actions, which may be a preferable choice to boost action recognition. For this reason, we propose a novel temporal memory module to hallucinate temporal features and make temporal predictions.

3.2.1 Temporal Hallucinating

Temporal hallucinating aims at taking advantage of video memory to learn temporal features of still images, i.e.,

$$\mathbf{U}^{flow}_{all} = \{\mathbf{u}^{flow}_{*},\ \mathbf{U}^{flow}\}. \tag{3}$$

In this work, we propose to adapt Gaussian process (GP) [24], a flexible Bayesian non-parametric model, to achieve this goal. The main reason is that, compared to parametric models, non-parametric models are often more suitable to match new examples from memory, without catastrophic forgetting [34]. To reduce writing redundancy, we mainly explain how to adapt GP for hallucination. More basics of GP can be found in our supplementary material and [24].

Spatial Similarity Comparison. As shown in Fig. 2, we first compare spatial similarities between images and videos, in order to weight which videos may be more relevant to each image. This can be elegantly achieved via the kernel operation of GP,

$$\mathbf{W}_h = \mathbf{K}_h(\mathbf{U}^{rgb}_{all},\ \mathbf{V}^{rgb})[\mathbf{K}_h(\mathbf{V}^{rgb},\ \mathbf{V}^{rgb}) + \sigma^2_h\mathbf{I}]^{-1} \tag{4}$$

where $\mathbf{U}^{rgb}_{all}$ refers to spatial features of query and training images in Eq. (2), $\mathbf{V}^{rgb}$ refers to spatial features of videos in Eq. (1), each entry of all kernel matrices $\mathbf{K}_h$ is computed by a kernel function $k_h(\mathbf{x}_i, \mathbf{x}_j)$ with two features $\mathbf{x}_i$ and $\mathbf{x}_j$, σ^2_h is a noise term, and each row of $\mathbf{W}_h$ is the weight vector of videos for an image.

Weighted Sum over Temporal Features of Videos. After obtaining the similarity matrix $\mathbf{W}_h$ between videos and images, we hallucinate temporal features of images via weighted sum over temporal features of videos,

$$\mathbf{U}^{flow}_{all} = \mathbf{W}_h\mathbf{V}^{flow}. \tag{5}$$

Note that, *temporal hallucinating is performed in an unsupervised manner, without using any labels information*. Hence, it can be used in the realistic scenarios, where image and video categories may be inconsistent.

3.2.2 Temporal Predicting

Once temporal features of still images are generated, we can make temporal prediction for query image. To achieve this goal, we adapt GP as follows.

Temporal Similarity Comparison. Since we not only have temporal features of videos $\mathbf{V}^{flow}$ but also obtain temporal features of few training images $\mathbf{U}^{flow}$, we integrate them together as temporal memory for query image, i.e.,

$$\mathbf{M}^{flow} = \{\mathbf{V}^{flow},\ \mathbf{U}^{flow}\}. \tag{6}$$

Subsequently, we compare temporal similarities between query image and this temporal memory, to see which videos

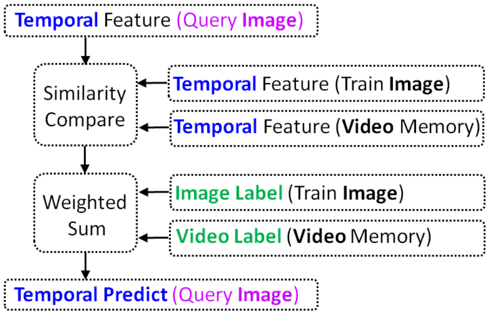
Temporal Feature (Query Image)
Similarity Compare
Temporal Feature (Train Image)
Temporal Feature (Video Memory)
Weighted Sum
Image Label (Train Image)
Video Label (Video Memory)
Temporal Predict (Query Image)

Figure 3. Temporal predicting in temporal memory module (better view in color). Specifically, we integrate temporal features of videos and training images as temporal memory for query image. Subsequently, we compare temporal similarities between query image and this temporal memory in a domain adaptation manner. Then, we compute the weighted sum over labels of temporal memory to obtain temporal prediction for query image. Note that, the structure of spatial memory module is the same as temporal predicting, except that all temporal terms are switched to be spatial. More explanations can be found in Section 3.2.2 and 3.3.

and training images are temporally relevant to the query image. Note that, the property of temporal similarities between query image and training images may be different from the one between query image and videos, due to shifts between image and video domains. For this reason, we perform the kernel operation of GP with a domain-adaptation noise term $\Sigma_p = \begin{bmatrix} \sigma_v^2\mathbf{I} & \mathbf{0} \\ \mathbf{0} & \sigma_u^2\mathbf{I} \end{bmatrix}$,

$$\mathbf{w}_p = \mathbf{K}_p(\mathbf{u}_*^{flow},\ \mathbf{M}^{flow})[\mathbf{K}_p(\mathbf{M}^{flow},\ \mathbf{M}^{flow}) + \Sigma_p]^{-1} \tag{7}$$

where $\mathbf{u}_*^{flow}$ is the temporal feature of query image that is obtained from temporal hallucinating, $\mathbf{M}^{flow}$ is the temporal memory in Eq. (6), each entry of all kernel matrices $\mathbf{K}_p$ is computed by a kernel function $k_p(\mathbf{x}_i, \mathbf{x}_j)$, σ_v^2 and σ_u^2 are respectively noise terms for videos and training images in temporal memory, and $\mathbf{w}_p$ is the weight vector of temporal memory for query image.

Weighted Sum over Labels of Temporal Memory. After obtaining $\mathbf{w}_p$ for query image, we make temporal prediction via weighted sum over labels of temporal memory,

$$\mathbf{L}_*^{flow} = \mathbf{w}_p\mathbf{L}, \tag{8}$$

where $\mathbf{L} = \{\mathbf{L}_v, \mathbf{L}_u\}$ refers to action labels of videos and training images in temporal memory. Since action categories may be different for images and videos, we use the total number of action categories in image and video domains to construct each one-hot-label vector in $\mathbf{L}$. As a result, *one can adaptively leverage videos and training images in temporal memory to boost temporal action prediction of query image.*

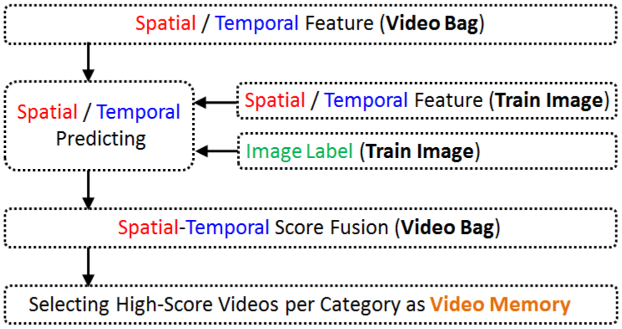
Spatial / Temporal Feature (Video Bag)
Spatial / Temporal Predicting
Spatial / Temporal Feature (Train Image)
Image Label (Train Image)
Spatial-Temporal Score Fusion (Video Bag)
Selecting High-Score Videos per Category as Video Memory

Figure 4. Video selection (better view in color). The goal is to mine highly-relevant videos from a video bag, so that we balance videos and training images to reduce inference bias as well as preserve computation efficiency. More details can be found in Section 3.4.

3.3. Spatial Memory Module

Since both image and video domains have spatial features, we introduce a spatial memory module for query image, where we integrate spatial features of videos and training images as spatial memory. Specifically, the memory structure of spatial predicting is the same as the one of temporal predicting in Section 3.2.2, except that all temporal terms are changed to be spatial. Finally, we perform spatial-temporal fusion as our final prediction for query image, due to the fact that spatial and temporal characteristics of actions are often complementary. Via designing temporal and spatial memory modules above, we flexibly leverage video memory to boost action recognition with few still images.

3.4. Video Selection Module

In practice, it is unnecessary and inefficient to use the entire video data set as memory, since not all videos are relevant to still images on hand. For this reason, we design a video selection module to mine highly-relevant videos $\{\mathbf{V}^{rgb},\ \mathbf{V}^{flow}\}$ from a video bag $\{\mathbf{V}_{bag}^{rgb},\ \mathbf{V}_{bag}^{flow}\}$, where this bag is collected by randomly sampling videos from each video-domain action category.

Step1: We use video bag as memory to hallucinate temporal features of training images. Hence, each image can be treated as a pseudo video with spatial and temporal features.

Step2: We use training images (i.e., pseudo videos in Step1) as memory, and perform spatial and temporal predicting for video bag. The prediction score for each video in the bag is about image-domain label.

Step3: We perform spatial and temporal score fusion for video bag, where each video has a fused score vector $\mathbf{s}_{fuse}$. We use $s_{max} = \max(\mathbf{s}_{fuse})$ as the importance of each video to image domain.

Step4: For each video-domain category in the bag, we select top N_{fuse} videos according to their s_{max}. Then, we use their features $\{\mathbf{V}^{rgb},\ \mathbf{V}^{flow}\}$ as our video memory. Note that, N_{fuse} is chosen to be the same as the number of training images in each image-domain category. In this case, we can balance videos and training images to reduce inference bias as well as preserve computation complexity.

4. Experiments

In this section, we evaluate our hybrid video memory (HVM) machine. To achieve this goal, we first introduce video and image data sets. More data explanations and experiments can be found in the supplementary material.

Data Sets. **(I) Video Memory**. We choose UCF101 (i.e., training set of split1) [30] as video memory of HVM, because it is a widely-used benchmark for action recognition in videos [28, 36, 37]. **(II) Still Image**. We use three still image data sets, i.e., WEB101, VOC, and DIFF20. First, we collect WEB101 and DIFF20 from internet, where WEB101 consists of the same 101 action categories as UCF101, and DIFF20 consists of 20 action categories that are different from the ones in UCF101. The action definition of DIFF20 can be found in our supplementary material. Second, VOC is built from VOC 2012 Action Dataset [6]. It consists of 10 action categories in which 4 categories are overlapped with UCF101 (i.e., Jumping, PlayingInstrument, RidingBike, RidingHorse). To avoid ambiguity of actions from multiple targets, we crop the squared bounding box as one image sample in our VOC. Furthermore, we exclude all samples in the 'other' class of VOC Action 2012 in our experiments, as our main goal is to evaluate if temporal features hallucinated from video can boost action recognition with few still images. Finally, the number of test images in WEB101/DIFF20 is 5,032/1,000 (around 50 test images per action category), and the number of test images in VOC is 2,658. Note that, we choose these data sets, since we aim at evaluating our HVM, when action difference between image and video domains is gradually increasing. We use the published protocol [28, 36, 37] to report classification accuracy for all the data sets.

Implementation Details. Unless stated otherwise, we perform our HVM machine with the following implementation details. First, we choose a widely-used two-stream CNN, i.e., Temporal Segment Net (TSN) [37], to obtain action representations of videos and images. Specifically, we use the published TSN that is trained on UCF101 split1 (i.e., our video memory). The spatial and temporal features are respectively generated from the 5b layer of two streams (1,024 dimension vector after global pooling). All deep features are then processed with l2 normalization and zero-mean operation. Second, the kernel of GP is chosen as the popular linear kernel (i.e., dot product), and all the hyper-parameters are carefully initialized. Third, we collect 1/5/10 training images from each action category of image sets to show the performance of HVM. Finally, we randomly select 50 videos from each action category of UCF101 (i.e., training set of split1), and pick the middle frame of each video to construct the video bag. Our video selection module picks the top 1/5/10 videos (per category) from this bag, and use them as our video memory.

4.1. Properties of Our HVM Machine

In this section, we mainly investigate different properties in HVM. To be fair, when we explore different strategies of one property, all other properties follow the basic strategy in the implementation details.

Key Modules of HVM. We evaluate the key modules of HVM in Table 1, where the baseline for comparison is that, we use spatial features of training images to perform the standard GP, without considering the proposed hybrid video memory structure. Additionally, the 'rand' setting is to randomly select 1/5/10 videos (per category from our video bag) as video memory, while the 'our' setting is to use the proposed video selection module to automatically select 1/5/10 videos (per category from our video bag) as video memory. **(I) Temporal and Spatial Memory Modules**. We examine the classification accuracy of temporal predicting (TP) in temporal memory module, spatial predicting (SP) in spatial memory module, and HVM (spatial-temporal fusion). First, 'our' TP and SP consistently outperform the baseline, especially when the training set is scarce (such as 1 image per category). It illustrates that, temporal and spatial memory modules can effectively take advantage of video memory to boost action recognition with few still images. Second, 'our' HVM achieves the best accuracy by fusing SP and TP, showing that spatial and temporal memory modules of HVM are complementary. Third, 'our' HVM works well for all data sets, even though action difference between image and video domains is increasing. It shows that the domain adaptation design in temporal and spatial predicting of HVM can successfully reduce the negative influence of domain difference and action difference. **(II) Video Selection Module**. First, the 'our' setting consistently outperforms the 'rand' setting, demonstrating that our video selection module learns a strongly-relevant video memory for still images. Second, the 'rand' TP is even worse than baseline for the 1-image setting in VOC and DIFF20. It illustrates that the 'rand' setting is prone to have more negative influence on temporal predicting, compared to spatial predicting. The main reason is that, the randomly-selected video memory may deteriorate temporal hallucinating of still images, and consequently reduce the performance of temporal predicting. This phenomenon is getting worse, when the difference of action categories between video and image is larger (such as VOC and DIFF20), and the number of training images is smaller (such as 1-image case). On the contrary, our video selection module adaptively finds a powerful video memory, which can boost both spatial and temporal predicting with few still images.

Different Model Choices. We evaluate two important

Data sets	WEB101			VOC			DIFF20		
No. of images	1-image	5-image	10-image	1-image	5-image	10-image	1-image	5-image	10-image
Baseline	26.6	48.6	57.3	39.0	59.1	64.7	57.4	81.9	84.5
rand TP	30.9	49.3	57.5	38.2	58.9	64.8	53.9	81.4	84.9
rand SP	31.5	50.7	58.5	39.0	59.4	65.7	58.0	83.2	84.8
rand HVM	33.1	51.3	58.6	39.3	59.7	66.0	59.1	83.4	85.7
our TP	33.2	50.5	57.6	42.0	59.1	65.1	59.1	82.5	85.5
our SP	33.0	51.3	58.8	39.3	59.7	66.3	58.3	83.4	85.2
our HVM	**35.4**	**52.3**	**59.2**	**42.2**	**60.1**	**66.5**	**60.2**	**83.5**	**86.4**

Table 1. Key modules of our HVM. Baseline: we perform the standard GP on the training images, without considering the proposed hybrid video memory structure. 'rand': we randomly select 1/5/10 videos (per category from video bag) as video memory. 'our': we use our video selection module to automatically select 1/5/10 videos (per category from video bag) as video memory. One can see that 'our' HVM achieves the best accuracy. More explanations can be found in the text.

Kernel Choice	WEB101			VOC			DIFF20		
of HVM	1-image	5-image	10-image	1-image	5-image	10-image	1-image	5-image	10-image
linKer	35.4	52.3	59.2	**42.2**	60.1	66.5	60.2	83.5	**86.4**
nonlinKer	**37.5**	**53.1**	**60.8**	41.2	**60.4**	**66.7**	**60.8**	**83.7**	86.3

Table 2. Different choices of kernel functions in HVM. linKer/nonlinKer: The kernel is linear/nonlinear in HVM, where linKer is the linear kernel (i.e., dot product), and nonlinKer is a popular neural network kernel in [24]. Our HVM with linear kernel is competitive to the one with nonlinear kernel. It illustrates that HVM is robust to different kernel choices.

model choices of HVM. **(I) Choices of Deep Action Representations**. We examine the performance of HVM, according to different choices of action representations. Here we choose the widely-used two-stream CNN architectures for comparison, i.e., Towards Good-Practice Net (TGPN) in [36] and Temporal Segment Net (TSN) in [37]. Both deep models are pre-trained on UCF101 (i.e., our video memory), where spatial and temporal features for TGPN are generated from the conv5_3 layer with global pooling (512 dimension vector), and the features for TSN are the same as before. As shown in Fig. 5, HVM achieves better accuracy with action representations from TSN. This is mainly because that, TSN is a deeper two-stream CNN which generates more discriminative features than TGPN. **(II) Choices of Kernel Functions**. We explore our HVM machine with different linear or nonlinear kernel functions. The linear kernel is the same as before, while the nonlinear kernel is a popular neural network kernel in [24]. In Table 2, HVM with linear kernel is competitive to the one with nonlinear kernel. This indicates that our HVM machine is robust to different kernel choices. For consistency, we use TSN and linear kernel in all our experiments.

Exploratory Analysis on HVM. We further analyze which types of action categories can be improved by HVM, compared to the baseline. For this reason, we choose the most challenging training setting (1-image per category), and compute the accuracy difference (%) between baseline and our memory modules in Fig. 6. As expected, temporal (or spatial) memory module is helpful to improve the motion (or appearance) related action categories, and HVM can leverage both spatial and temporal features to enhance the final prediction. Furthermore, the improvement of temporal memory module is generally larger than the one of spatial memory module. It illustrates that, the hallucinated temporal features can generate more discriminative motion characteristics for still images, especially when the training set is scarce. Finally, we investigate the largest-improved action category by HVM, i.e., BlowingCandles / RidingHorse / AmericanFootball for WEB101 / VOC / DIFF20. The results in Table 3 show that, the mistakes introduced by the most confused category (i.e., BrushingTeeth / RidingBike / PlayingHandball) can be significantly decreased by HVM. Hence, our HVM machine can effectively leverage video memory for action recognition with few still images.

4.2. Comparison with Related Works

We compare our HVM machine with a number of related works. Since temporal hallucinating is designed to boost action recognition with few still images, we choose the most challenging 1-image case to show the effectiveness. Specifically, we categorize the related works into three groups. **(I)** Traditional classifiers, i.e., K Nearest Neighbors (KNN) and SVM. Since 1 training image is available for each action category, we choose the nearest neighbor (K=1) in KNN. All the hyper parameters of SVM are selected by grid search algorithm in LIBSVM. We train both approaches with deep features of training images. **(II)** Transfer learning with deep models in action recognition. Following the transfer strategy in [40], we fine-tune the last layer of popular deep models in action recognition, i.e., TGPN [36], TSN [37] and

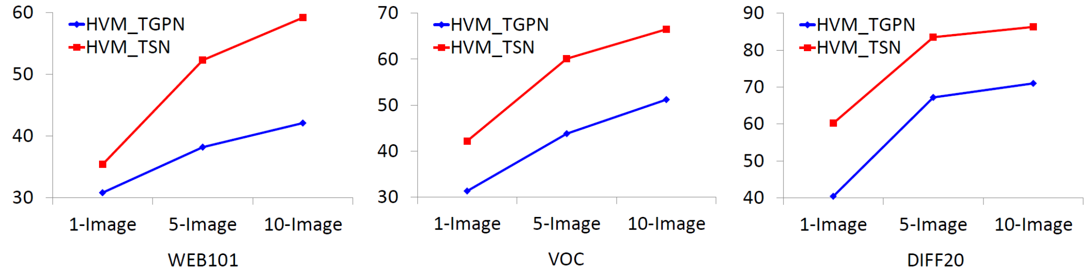

Figure 5. Different choices of deep action representations in our HVM. We examine our HVM, according to action representations of different two-stream CNN architectures, i.e., Towards Good-Practice Net (TGPN) in [36] and Temporal Segment Net (TSN) in [37]. Our HVM achieves better accuracy with action representations from TSN. This is mainly because that, TSN is a deeper two-stream CNN which generates more discriminative features than TGPN.

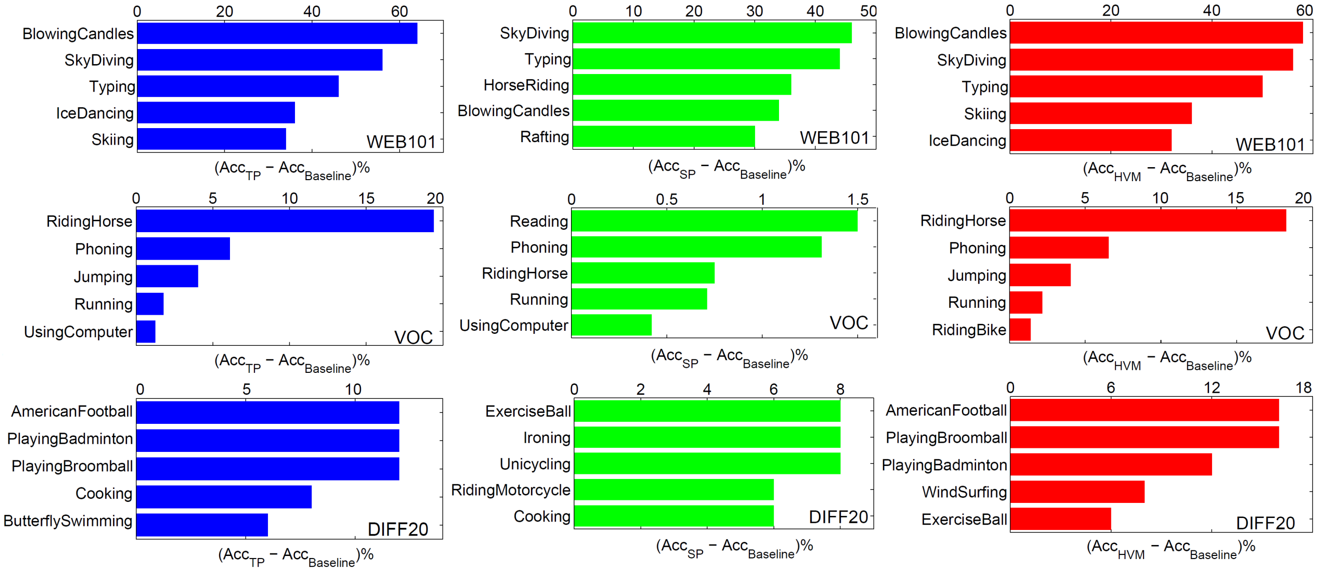

Figure 6. Exploratory action analysis of HVM. We choose the most challenging training setting (1-image per category), and compute the accuracy difference (%) between baseline and our memory modules. First, temporal (or spatial) memory module is helpful to improve the motion (or appearance) related categories, and our HVM can leverage both spatial and temporal features to enhance the final prediction. Second, the improvement of temporal memory module is generally larger than the one of spatial memory module. It illustrates that, the hallucinated temporal features can generate more discriminative characteristics for still images.

R*CNN [8]. All deep models are pre-trained on UCF101 (i.e., our video memory). Additionally, R*CNN requires ground truth bounding boxes of human actions in the image set. It can be an expensive annotation procedure, which is beyond our goal in this paper. Hence, we show its result on VOC, where the bounding boxes are available. **(III)** Well-known memory machines in deep learning community, i.e., KV-MemNNs [21] and Matching Network [34]. For KV-MemNNs, we use spatial video memory as key and temporal video memory as value. Similar to our HVM machine, its final prediction for query is based on both key and value. For Matching Network, LSTM is trained with spatial features of video bag, where the number of randomly-sampled action categories (per minibatch) is the same as the one of action categories in the target image sets. The results are shown in Table 4. First, our HVM outperforms the traditional classifiers (KNN and SVM) and deep transferred models (TGPN, TSN and R*CNN), showing the effectiveness of video memory. Second, our HVM outperforms the recent KV-MemNNs and Matching Network. It demonstrates that our HVM can be a preferable video memory network to boost action recognition with few images.

4.3. Visualization

An important merit of our HVM machine is temporal hallucinating in the temporal memory module, since this procedure can unsupervisedly generate temporal features that are originally non-existent for still images. After evaluating it quantitatively in the previous sections, we choose DIFF20 to visualize temporal hallucinating. This choice is based on the fact that, action categories of DIFF20 are different from our video memory (i.e., UCF101). Using this

Action	WEB101	VOC	DIFF20
Confusion	BlowingCandles→BrushingTeeth	RidingHorse→RidingBike	AmericanFootball→PlayingHandball
Baseline	20 mistaken images	103 mistaken images	20 mistaken images
Our HVM	**6** mistaken images	**62** mistaken images	**11** mistaken images

Table 3. Confusion Reduction by HVM. The largest-improved action category in HVM is BlowingCandles / RidingHorse / American-Football for WEB101 / VOC / DIFF20, where the mistakes introduced by the most confused category (i.e., BrushingTeeth / RidingBike / PlayingHandball) can be significantly decreased by our HVM machine.

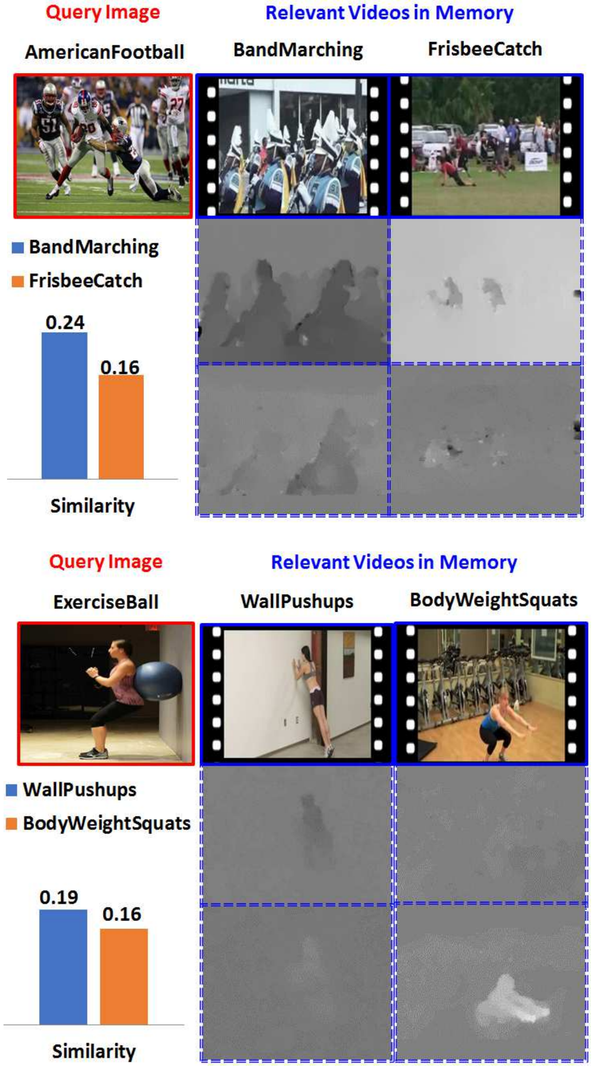

Figure 7. Visualization of temporal hallucinating (DIFF20 with 1-image case). Temporal hallucinating correctly compares similarities between query images and videos in memory, and unsupervisedly generates temporal features with motion cues for still images.

data can further confirm the unsupervised learning benefits of our temporal hallucinating. The results for the most challenging 1-image setting are shown in Fig. 7, where we demonstrate the query images (RGB) and two most relevant videos (RGB and optical flows) in our video memory.

Approaches	WEB101	VOC	DIFF20
KNN	26.1	38.3	55.7
SVM	22.3	32.0	54.2
TGPN [36]	15.5	30.5	35.2
TSN [37]	26.1	40.3	56.3
R*CNN [8]	n/a	28.3	n/a
KV-MemNNs [21]	24.4	39.5	52.1
Matching Network [34]	26.6	39.9	56.7
Our HVM	**35.4**	**42.2**	**60.2**

Table 4. Comparison with related works (most challenging 1-image case). More explanations can be found in the text.

The relevant videos in memory is obtained by computing the weight vector of temporal hallucinating in Eq. (4). One can clearly see that, temporal hallucinating can correctly compare similarities between a query image and videos in memory, and unsupervisedly generate temporal features with motion characteristics for still images.

5. Conclusion

In this paper, we propose a novel hybrid video memory (HVM) machine to boost action recognition with few training images. First, temporal memory module can unsupervisedly hallucinate temporal features of still images from video memory, and effectively make temporal prediction for query images, with consideration of domain difference between images and videos. Second, spatial memory module can make spatial prediction of query images. Due to complementary properties of spatial and temporal features, we apply spatial-temporal prediction fusion to further enhance the performance. Finally, video selection module can select the strongly-relevant videos as memory, which can reduce prediction bias while preserving computation efficiency. In our experiments, HVM outperforms a number of recent works, showing that it is a preferable video memory machine for action recognition with few images.

Acknowledgements. This work was supported in part by National Key Research and Development Program of China (2016YFC1400704), National Natural Science Foundation of China (61502470, U1613211, 61633021), Shenzhen Research Program (JCYJ20160229193541167), and Industrial Collaboration Project (Y5Z0371001).

References

[1] L. Bertinetto, J. F. Henriques, J. Valmadre, P. H. S. Torr, and A. Vedaldi. Learning feed-forward one-shot learners. In *NIPS*, 2016.

[2] A. Diba, A. M. Pazandeh, H. Pirsiavash, and L. V. Gool. Deepcamp: Deep convolutional action & attribute mid-level patterns. In *CVPR*, 2016.

[3] A. Dosovitskiy, P. Fischer, E. Ilg, P. Häusser, C. Hazirbaş, V. Golkov, P. van der Smagt, D. Cremers, and T. Brox. FlowNet: Learning Optical Flow with Convolutional Networks. In *ICCV*, 2015.

[4] W. Du, Y. Wang, and Y. Qiao. Rpan: An end-to-end recurrent pose-attention network for action recognition in videos. In *ICCV*, 2017.

[5] W. Du, Y. Wang, and Y. Qiao. Recurrent spatial-temporal attention network for action recognition in videos. *IEEE TIP*, 2018.

[6] M. Everingham, L. Gool, C. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *IJCV*, 2010.

[7] G. Gkioxari, R. Girshick, and J. Malik. Actions and attributes from wholes and parts. In *ICCV*, 2015.

[8] G. Gkioxari, R. Girshick, and J. Malik. Contextual action recognition with r*cnn. In *ICCV*, 2015.

[9] A. Graves, G. Wayne, and I. Danihelka. Neural Turing Machines. In *arXiv:1410.5401*, 2014.

[10] G. Guo and A. Lai. A survey on still image based human action recognition. *Pattern Recognition*, 2014.

[11] B. Hariharan and R. B. Girshick. Low-shot visual object recognition. *arXiv:1606.02819*, 2016.

[12] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *arXiv:1512.03385*, 2015.

[13] F. C. Heilbron, V. Escorcia, B. Ghanem, and J. C. Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *CVPR*, 2015.

[14] L. Kaiser, O. Nachum, A. Roy, and S. Bengio. Learning to Remember Rare Events. In *ICLR*, 2017.

[15] 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.

[16] G. Koch, R. Zemel, and R. Salakhutdinov. Siamese Neural Networks for One-shot Image Recognition. In *ICML Workshop*, 2015.

[17] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*. 2012.

[18] S. Kwak, M. Cho, and I. Laptev. Thin-slicing for pose: Learning to understand pose without explicit pose estimation. In *CVPR*, 2016.

[19] B. M. Lake, R. Salakhutdinov, and J. B. Tenenbaum. Human-level concept learning through probabilistic program induction. *Science*, 2015.

[20] X. Liang, L. Lee, W. Dai, and E. P. Xing. Dual motion gan for future-flow embedded video prediction. In *ICCV*, 2017.

[21] A. H. Miller, A. Fisch, J. Dodge, A.-H. Karimi, A. Bordes, and J. Weston. Key-Value Memory Networks for Directly Reading Documents. In *EMNLP*, 2016.

[22] J. Y. Ng, M. J. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and G. Toderici. Beyond short snippets: Deep networks for video classification. *CoRR*, 2015.

[23] J. Y.-H. Ng, J. Choi, J. Neumann, and L. S. Davis. ActionFlowNet: Learning Motion Representation for Action Recognition. In *arXiv:1612.03052*, 2017.

[24] C. E. Rasmussen and C. K. I. Williams. *Gaussian Process for Machine learning*. MIT Press, 2006.

[25] S. Ravi and H. Larochelle. Optimization as a model for few-shot learning. In *ICLR*, 2017.

[26] Z. Ren, J. Yan, B. Ni, B. Liu, X. Yang, and H. Zha. Unsupervised Deep Learning for Optical Flow Estimation. In *AAAI*, 2017.

[27] A. Santoro, S. Bartunov, M. Botvinick, D. Wierstra, and T. Lillicrap. One-shot Learning with Memory-Augmented Neural Networks. In *ICML*, 2016.

[28] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. *CoRR*, 2014.

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

[30] K. Soomro, A. R. Zamir, and M. Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv:1212.0402*, 2012.

[31] N. Srivastava, E. Mansimov, and R. Salakhutdinov. Unsupervised Learning of Video Representations using LSTMs. *ICML*, 2015.

[32] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *CVPR*, 2015.

[33] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3d convolutional networks. In *ICCV*, 2015.

[34] O. Vinyals, C. Blundell, T. Lillicrap, K. Kavukcuoglu, and D. Wierstra. Matching networks for one shot learning. In *NIPS*, 2016.

[35] C. Vondrick and A. Torralba. Generating the future with adversarial transformers. In *CVPR*, 2017.

[36] L. Wang, Y. Xiong, Z. Wang, and Y. Qiao. Towards good practices for very deep two-stream convnets. *CoRR*, 2015.

[37] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. V. Gool. Temporal segment networks: Towards good practices for deep action recognition. In *ECCV*, 2016.

[38] Z. Wang, L. Wang, Y. Wang, B. Zhang, and Y. Qiao. Weakly supervised patchnets: Describing and aggregating local patches for scene recognition. *IEEE TIP*, 2017.

[39] Z. Xu, L. Zhu, and Y. Yang. Few-shot object recognition from machine-labeled web images. *arXiv:1612.06152*, 2016.

[40] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson. How transferable are features in deep neural networks? In *NIPS*. 2014.

[41] Y. Zhang, L. Cheng, J. Wu, J. Cai, M. N. Do, and J. Lu. Action recognition in still images with minimum annotation efforts. *IEEE TIP*, 2016.

[42] Y. Zhu, Z. Lan, S. Newsam, and A. G. Hauptmann. Hidden Two-Stream Convolutional Networks for Action Recognition. In *arXiv:1704.00389*, 2017.