Spatiotemporal Pyramid Network for Video Action Recognition[*]

Yunbo Wang[†], Mingsheng Long[†], Jianmin Wang[†], and Philip S. Yu[†‡]

[†]KLiss, MOE; TNList; NEL-BDSS; School of Software, Tsinghua University, China

[‡]University of Illinois at Chicago, IL, USA

wangyb15@mails.tsinghua.edu.cn, {mingsheng,jimwang}@tsinghua.edu.cn, psyu@uic.edu

Abstract

Two-stream convolutional networks have shown strong performance in video action recognition tasks. The key idea is to learn spatiotemporal features by fusing convolutional networks spatially and temporally. However, it remains unclear how to model the correlations between the spatial and temporal structures at multiple abstraction levels. First, the spatial stream tends to fail if two videos share similar backgrounds. Second, the temporal stream may be fooled if two actions resemble in short snippets, though appear to be distinct in the long term. We propose a novel spatiotemporal pyramid network to fuse the spatial and temporal features in a pyramid structure such that they can reinforce each other. From the architecture perspective, our network constitutes hierarchical fusion strategies which can be trained as a whole using a unified spatiotemporal loss. A series of ablation experiments support the importance of each fusion strategy. From the technical perspective, we introduce the spatiotemporal compact bilinear operator into video analysis tasks. This operator enables efficient training of bilinear fusion operations which can capture full interactions between the spatial and temporal features. Our final network achieves state-of-the-art results on standard video datasets.

1. Introduction

Learning a good video representation is the foundation of many computer vision tasks, such as action recognition and video captioning. It goes beyond image analysis and depends on a joint modeling of both spatial and temporal cues. Many existing methods [12, 10, 28, 26] are dedicated to this modeling by taking advantages of Convolutional Neural Networks (CNN) [16, 13, 23, 27]. However, these CNN-based methods have not shown an overwhelming performance over other approaches [30, 31] using Fisher Vector [19], HOF [15], and dense trajectories [30]. One reason is that these CNN frameworks are not specifically designed for videos and cannot fully exploit spatiotemporal features.
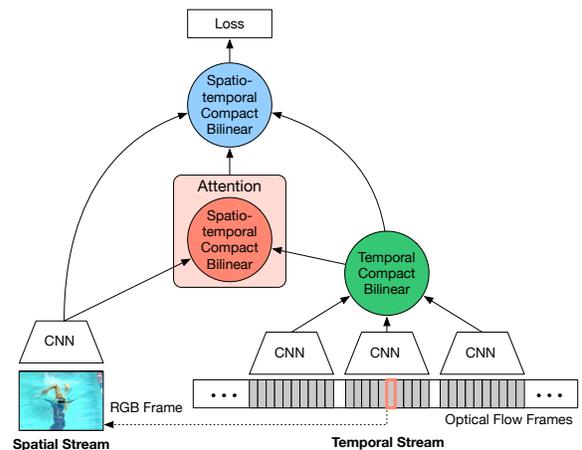
Figure 1. An overview of our spatiotemporal pyramid network, which constitutes a multi-level fusion pyramid of spatial features, long-term temporal features and spatiotemporal attended features.

In addition to capturing the appearance information using standard CNN stream, several recent approaches try out using optical flow data in a second CNN stream to capture the motion information [22, 37, 6, 38]. However, when taking a closer look at these models, we observe that for most misclassification cases, there is usually one stream failing, while the other remaining correct. Hence, simply averaging the outputs of the classifier layers is not enough. Instead, we hope to make the spatial and temporal cues facilitate each other. This paper presents a novel end-to-end spatiotemporal pyramid architecture, as shown in Figures 1, which can on one hand boost the accuracy of individual stream and on the other hand exploits spatiotemporal cues jointly.

From the temporal perspective, since the original optical flow stream only receives 10 consecutive optical flow frames, it may be fooled if two actions resemble in such a short snippet, though distinguish in the long term. A typical example is Pull-ups and RopeClimbing in UCF101: men in these two categories could be moving in the same single direction in short snippets. But if we enlarge the time range, we can easily find that the man in Pull-ups is actually moving up and down, while the one in RopeClimbing is mov-

---

[*]Contact author: Mingsheng Long (mingsheng@tsinghua.edu.cn).

ing straight upwards. To learn more global video features, we use multi-path temporal subnetworks to sample optical flow frames in a longer sequence, and explore several fusion strategies to combine the temporal information effectively.

From the spatial perspective, the original spatial stream is easily fooled when the backgrounds of two videos are extremely similar. For instance, it cannot tell FrontCrawl and BreastStroke apart, because for both categories, the swimming pool turns out to be the strongest feature. However, the optical flow network can tell these actions quite clearly and outperform the spatial stream by at least 5% on UCF101. Motivated by this, we introduce a spatiotemporal attention module to extract significant locations on feature maps of the spatial network. In this process, the temporal features are exploited as a guidance, which informs the spatial stream where the motion of interest happens.

Besides improving the effectiveness of individual streams, we explore methods to fuse the spatiotemporal features, which enable a joint optimization of the whole architecture using a unified spatiotemporal loss function. We bring in the compact bilinear fusion strategy, which captures full interactions across spatial and temporal features, while significantly reduces the number of parameters of traditional bilinear fusion methods from millions to just several thousands. Our experiment results demonstrate that compact bilinear approximately models the correlations between each single element of the spatial and temporal features, thus yields better performance over its alternatives, such as concatenation and element-wise sum studied in [6].

Our contributions can be summarized as follows. **(a)** We propose a novel deep learning architecture to address the problems we observe in video representation learning. **(b)** We introduce the compact bilinear and spatiotemporal attention methods into video-related tasks and validate their feasibility in practice. **(c)** We evaluate our approaches on standard video datasets UCF101 and HMDB51 and report significant improvement over the previous state-of-the-art.

2. Related Work

Motivated by the impressive performance of deep learning on image-related tasks, several recent works try to design effective CNN-based architectures for video recognition that jointly model spatial and temporal cues. Before the blossom of CNN, Ji et al. [10] first exploit consecutive video frames as inputs and extend the convolutional filters into temporal domain. Karpathy et al. [12] compare multiple CNN connectivity methods in time, including late fusion, early fusion and slow fusion. But these approaches cannot fully utilize motion information, and only yield a modest improvement over operating on single frames. Tran et al. [28] train a deeper CNN model called C3D on Sports-1M. C3D is basically a 3D version of VGGnet [23], containing 3D convolution filters and 3D pooling layers op-

	Ours	Two-Stream Fusion
Arch.	Pyramid	Single Conv-Layer
#Loss	1 (End-to-End)	2 (Average)
Spatial	Attention Pooling	Average Pooling
Temporal	Compact Bilinear	3D Max Pooling

Table 1. Differences between our method and state-of-the-art [6].

erating over space and time simultaneously. Noticing that stacked RGB frames cannot fully exploit temporal cues, Simonyan et al. [22] train a second stream of CNN on optical flow frames and propose a two-stream ensemble network. Since the optical flow data brings in a significant performance gain, it has recently been employed into many other action recognition methods [2, 5, 25, 29, 34, 37, 26, 33, 6].

However, the original two-stream method [22] has two main drawbacks: First, it only incorporate 10 consecutive optical flow frames, so that it cannot capture long-term temporal cues. Second, it cannot learn the subtle spatiotemporal relationships. The spatial (RGB frames) and temporal (optical flow) streams are trained separately, and the final predictions are obtained by averaging the outputs of two classifiers. To mitigate these issues, Ng et al. [37] investigate several pooling methods as well as the Long Short-Term Memory (LSTM) [37] to fuse features across a longer video sequence. Wang et al. [32] model long-term temporal structures by proposing a segmental network architecture with sparse sampling. Feichtenhofer et al. [6] study multiple ways of combining networks both spatially and temporally. They propose a spatiotemporal fusion method and claim that the two-stream networks should be fused at the last convolutional layer. Table 1 lists the main differences between our work and [6]. First and foremost, we propose a multi-layer pyramid fusion architecture, replacing a 3D convolutional layer and a pooling layer in [6], to combine the spatial and temporal features at different abstraction levels. For individual streams, we upgrade the spatial subnetwork by replacing the original average pooling with a spatiotemporal attention module. This method makes the network concentrate on significant regions on static frames with the help of motion cues. Moreover, we introduce the compact bilinear operator for fusing multi-path optical flow features temporally. Finally, in training strategy of [6], two losses are used in their objective function and the final prediction is obtained by averaging the outputs of two streams. In contrast, our fusion network is trained end-to-end with one single spatiotemporal loss function. Thus, all streams are optimized as a whole, resulting in an improved result.

3. Spatiotemporal Pyramid Network

The spatiotemporal pyramid network supports long-term temporal fusion and a visual attention mechanism. Also, we propose a new spatiotemporal compact bilinear operator to enable a unified modeling of various fusion strategies.

3.1. Spatiotemporal Compact Bilinear Fusion

The fusion of spatial and temporal features in compact representations proves to be the key to learning high-quality spatiotemporal features for video recognition. A good fusion strategy should maximally preserve the spatial and temporal information while maximize their interaction. Typical fusion methods including element-wise sum, concatenation, and bilinear fusion have been extensively evaluated in the convolutional two-stream fusion framework [6]. However, element-wise sum and concatenation do not capture the interactions across the spatial and temporal features, so they may suffer form substantial information loss. Bilinear fusion allows all spatial and temporal features in different dimensions to interact with each other in a multiplicative way. Since our spatiotemporal pyramid constitutes spatial features, temporal features, and their hierarchy, the bilinear fusion is the only appropriate strategy for our approach.

Specifically, denote by x and y the spatial and temporal feature vectors respectively, the bilinear fusion is defined as $z = \text{vec}(x \otimes y)$, where $\otimes$ denotes the outer product xy^T, and vec denotes the vectorization of a vector. Bilinear fusion leads to high dimensional representations with million of parameters, which will make network training infeasible.

To circumvent the curse of dimensionality, we propose a Spatiotemporal Compact Bilinear (STCB) operator to enable various fusion strategies. We transform the outer product to a lower-dimensional space which avoids computing the outer product directly. As suggested by the compact bilinear pooling method [7], for a single modality, we adopt the Count Sketch projection function Φ [1], which projects a vector $v \in \mathbb{R}^p$ to $v' \in \mathbb{R}^d$. We initialize two vectors $s \in \{-1, 1\}^p$ and $h \in \{1, \ldots, d\}^p$, where s contains either 1 or -1 for each index, and h maps each index j in the input v to an index k in the output v'. Both s and h are initialized randomly from a uniform distribution and remain constant for future invocations of Count Sketch. v' is initialized as a zero vector. For every entry $v(j)$ its destination index $k = h(j)$ is looked up by h, and $s(j) \cdot v(j)$ is added to $v'(k)$. See Algorithm 1 for the details, where m is the number of feature pathways for compact bilinear fusion.

This procedure enables projecting the outer product of spatial and/or temporal features into a lower-dimensional space, which significantly reduces the number of parameters from millions to several thousands. To avoid explicitly computing the outer product, [20] reveals that the Count Sketch of the outer product of two vectors can be expressed as convolution of both Count Sketches: $\psi(x \otimes y, h, s) = \psi(x, h, s) * \psi(y, h, s)$, where $*$ is the convolution operator. Fortunately, the convolution theorem states that convolution in the time domain is equivalent to element-wise product in the frequency domain. Thus the convolution $x * y$ can be rewritten as $\text{FFT}^{-1}(\text{FFT}(x) \odot \text{FFT}(y))$, where $\odot$ refers to element-wise product. These ideas are summarized in Al-gorithm 1, which is based on the Tensor Sketch algorithm [20]. We invoke the algorithm with m pathways of spatial and/or temporal features that need to be fused, which enables spatiotemporal fusion into compact representations.

Algorithm 1: STCB: Spatiotemporal compact bilinear

Input: Spatial and/or temporal features $\{v_i \in \mathbb{R}^{p_i}\}_{i=1}^m$
Output: Fused features $\Phi(\{v_i\}_{i=1}^m) \in \mathbb{R}^d$

1   **for** $i \leftarrow 1$ **to** m **do**
2   **if** h_i, s_i not initialized **then**
3   **for** $j \leftarrow 1$ **to** p_i **do**
4   sample $h_i(j)$ from $\{1, \ldots, d\}$
5   sample $s_i(j)$ from $\{-1, 1\}$
6   **end**
7       $v_i' = [0, \ldots, 0]$
8   **for** $j \leftarrow 1$ **to** p_i **do**
9         $v_i'(h_i(j)) = v_i'(h_i(j)) + s_i(j) \cdot v_i(j)$
10   **end**
11   **end**
12     $\Phi(\{v_i\}_{i=1}^m) = \text{FFT}^{-1}\left(\bigodot_{i=1}^m \text{FFT}(v_i')\right)$
13 **end**

3.2. Temporal Fusion

The original temporal stream takes 10 consecutive optical flow frames as inputs, thus it may make mistakes if two actions look similar in such short snippets, though differentiate in the long term. Therefore, we hypothesize that learning a more global representation would offer more accurate motion cues for the upper levels of the pyramid network. Specifically, we enlarge the input video chunks by using multiple CNNs with shared network parameters to sample the optical flow frames at an interval of τ. Each chunk covers the previous $L/2$ and the next $L/2$ frames as inputs. For both training and testing, L is fixed to 10, and τ is randomly selected from 1 to 10, in order to model variable lengths of videos with a fixed number of neurons. For the fusion method, we exploit STCB and make it support a scalable number of input feature maps. We show that STCB is effective not only for spatiotemporal fusion, but also for temporal combination.

Comparing our method with [6, 32], all these three methods aim to broaden the input fields across the time domain. Multiple snippets are fused by 3D pooling in [6] and by compact bilinear in our work. Another difference between our method and [6] is that their temporal fusion includes fusing the features of multiple RGB frames as well, while we only combine optical flow representations. The reason is that modeling multiple RGB frames with another loss function would cause additional overfitting issue in training and obtain misleading results in testing. More importantly, in [6, 32], the resulting temporal features are directly fed into classifiers. In this paper, they are used as input to the next
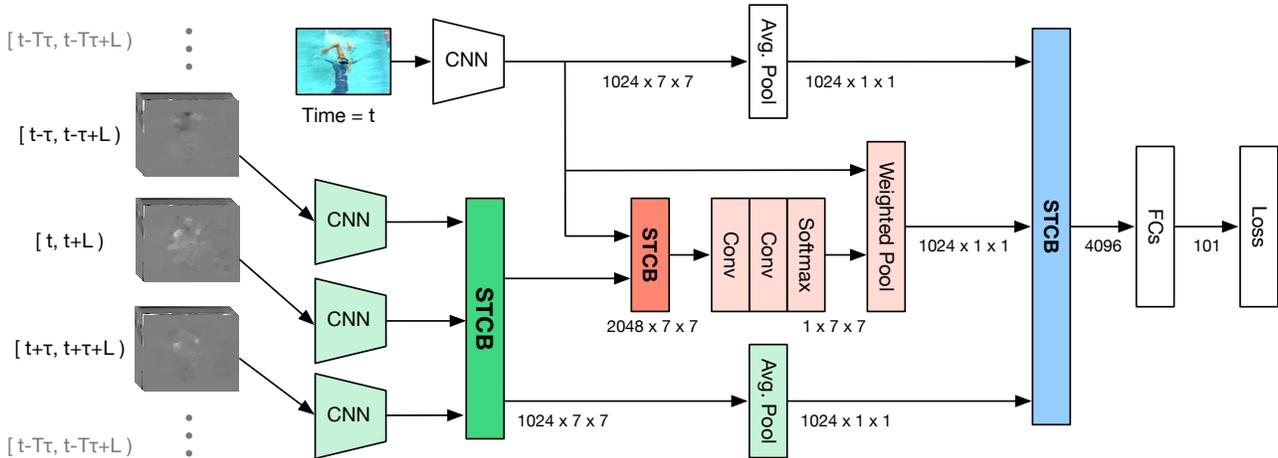
Figure 2. The spatiotemporal pyramid network learns spatiotemporal features at multiple abstraction levels, which can be trained end-to-end as a whole. Optical flow features are first combined across time by a compact bilinear layer. The resulting features then run through the second compact bilinear layer and a spatiotemporal attention module, determining the salient regions of activities. The final video representations are obtained by fusing features from the spatial stream, the temporal stream and the attention stream.

fusion stage (attention) in our architecture. We observe that compact bilinear fusion can preserve the temporal cues to supervise the spatiotemporal attention module.

3.3. Spatiotemporal Attention

The second level of our spatiotemporal fusion pyramid is a variant of the attention model, which is originally proposed in multi-modal tasks [36, 35, 18]. We adopt this idea and extensively apply it to the spatiotemporal scenario, by taking advantage of the motion information to locate salient regions on the image feature maps. We implement it on the last convolutional layers (i.e. inception5b in BN-Inception, res5c in ResNets and conv5 in VGGnet). For one thing, the representations of these layers show class-specific motion variations, while the lower layers capture finer-grained features of the image, such as edges, corners and textures. For another thing, we want the attention module to play a role as a more accurate weighted pooling operation, because we believe that the original average pooling cannot capture the salient regions corresponding to the activity information and may results in information loss.

The spatiotemporal attention module reduces a $1048 \times 7 \times 7$ feature map in BN-Inception to a $1048 \times 1 \times 1$ feature vector. But unlike max pooling or average pooling, the attention pooling has a relatively sophisticated logic and complicated mechanism. More specifically, for each grid location on the image feature maps, we use STCB to merge the spatial feature vector with its temporal counterpart. The output spatiotemporal representations, implying the corresponding appearance and motion cues, serve as the supervision of the following attention layers. After that, two convolutional layers are stacked to produce the attention weights

for the feature maps. The size of the first convolutional layer is $64 \times 7 \times 7$, while that of the second is $1 \times 7 \times 7$. At last, the resulting attention weights are normalized by a softmax layer, and then combined with the original spatial features by a weighted pooling layer. The spatiotemporal attention with STCB makes the spatial stream prone to be abstracted by the moving objects in the static RGB frames.

Though the attention mechanism has been explored in action recognition, our work differs from the others, such as [21] in two folds: **(a)** our attention is generated by both spatial-stream and temporal-stream CNNs while the others' are generated by LSTMs, both to highlight the motion cues in the spatial representations; **(b)** we further use optical flow features as the temporal supervision to the attention module.

3.4. Proposed Architecture

All techniques mentioned above can be integrated under a pyramid framework. We design our architecture by injecting the proposed fusion layers between the convolutional and the fully connected layers. Under these circumstances, we only use the representations of the last convolutional layer, so that our approach can be extendible for almost all CNN architectures, including BN-Inception, ResNets and VGGnet. As a result, we can initialize our network with models that are well pre-trained on ImageNet [4] before we fine-tune them on the relatively small video datasets.

We use the STCB technique three times. At the bottom of the pyramid, multiple optical flow representations across a longer video sequence are combined by the first STCB (green layers in Figure 2). By doing this, we obtain more global temporal features. These features are then fed into the next fusion level, the spatiotemporal attention subnet-

work (red layers), where we use another STCB to fuse the spatial feature maps with the corresponding motion representations, and offer the attention cues of salient activities. At the top of the fusion pyramid, all the three previous outcomes are used: the original spatial and temporal features through average pooling, as well as the resulting attended features through the attention module. They are fused into a 4096-dimensional vector by a third STCB, and each of them captures significant information of multiple abstraction levels in the pyramid. Removing any feature pathway would result in a decrease in the overall performance.

4. Experiments

This section is organized in accordance with the progress of our experiments. Initially, we describe the datasets and training details. Then, we explore the effects of applying different deep networks as the building blocks of our model, including VGGnet [23], ResNets [8] and BN-Inception [9]. Next, we evaluate our spatiotemporal pyramid network and demonstrate its effectiveness by giving ablation results. Last but not least, we compare our method with the previous state-of-the-art and analyze its performance by giving typical examples of correct and incorrect predictions.

4.1. Datasets and Implementations

We train and evaluate our spatiotemporal pyramid network on two standard datasets. The UCF101 dataset [24] contains 13320 fully-annotated video snippets from 101 action categories. Each snippet lasts 3-10 seconds and consists of 100-300 frames on average. The HMDB51 dataset [14] contains 6766 videos clips that covers 51 action categories. For both of them, we follow the provided evaluation protocol and adopt standard training/testing splits.

To verify the pure effectiveness of different pyramid fusion layers, we train a VGG-16 model, a BN-Inception model and a ResNet-50 model respectively on standard video datasets. Our models are trained following a multi-stage training strategy. We initialize the spatial and the temporal subnetworks with models pre-trained on ImageNet. Then we fine-tune each of them on the target video datasets and fill these parameters into our final pyramid network. We follow the cross modality fine-tuning strategy presented in [32]. After that, we train the entire network on UCF101 or HMDB51. Here we choose the mini-batch stochastic gradient descent algorithm and set the batch size to 32. Moreover, for VGG-16, we stack the two 4096-dimensional fully connected layers behind the last STCB layer. We set the base learning rate to 0.01 and decrease it by a factor of 10 every 10,000 iterations, and the training process stops at iteration 30,000. For ResNets, the base learning rate is initialized as 0.001, reduce by a factor of 10 every 10,000 iterations, and stopped at 20,000. To avoid overfitting, we randomly sample the temporal interval τ from 1 to 10. Also,

we exploit several data augmentation techniques, such as scale jittering, horizontal flipping and image cropping. Details of these tricks are not in the scope of this paper. All experiments are implemented with Caffe [11].

4.2. Base Architectures

Deeper CNNs can often lead to better performance in image recognition tasks [3, 17, 27, 8], since they bring in great modeling capacity and are capable of learning discriminative representation from raw visual data. The state-of-the-art two-stream architecture [6] is based on VGG-16 [23], while Inception with Batch Normalization (BN-Inception) [9] and Deep Residual Networks (ResNets) [8] have shown remarkable performance in several challenging recognition tasks recently. In this work we further explore the feasibility of ResNets in video analysis tasks. We take into account a 50-layer ResNet for the sake of computational complexity, as well as a 152-layer ResNet for its compelling accuracy. All models are pre-trained on the ImageNet [4] and fine-tuned on UCF101 and HMDB51.

Table 2 compares the performance of VGGnet, BN-Inception and ResNets. Generally, as the number of convolutional layers grows, the RGB network benefits most. In contrast, the performance of the optical flow network decreases slightly. There are two reasons. First, the optical flow data yields a different distribution from RGB, which weakens the impact of fine-tuning. Second, due to the limited amount of training samples on UCF101, complex network structures are prone to over-fitting. BN-Inception turns out to be the top-performing base architecture.

Model	Spatial	Temporal	Two-Stream [22]
VGG-16	80.5%	85.4%	88.9%
ResNet-50	83.7%	84.9%	90.3%
ResNet-152	84.3%	82.1%	89.8%
BN-Inception	84.5%	87.0%	91.7%

Table 2. Classification accuracy of the two-stream model [22] with different base architectures on UCF101 (Split 1). All results are obtained by averaging the outputs of the Softmax layers as [22].

4.3. Spatiotemporal Compact Bilinear Fusion

We explore several strategies for fusing spatial and temporal feature maps. All models but the VGGnet one follow the same architecture, that the fusion layer is put between the last convolutional layer (i.e. res5c for ResNets and inception5b for BN-Inception) and the final classifier. Our experiments show that such a late fusion architecture outperforms its alternatives in which the fusion layer is moved forward. It can be explained by that the last convolutional layer shows class-specific and highly informative features with significant motion variations.

As shown in Table 3, spatiotemporal compact bilinear

fusion results in the highest accuracy and improves the performance by around 1.5 points. It is a valuable observation. Before this, what we know is that compact bilinear pooling is effective for combining visual representations. But what we do not know is that the same merit happens between spatial and temporal data. Table 3 also reveals that the output dimension makes a difference on the performance of spatiotemporal compact bilinear fusion. As it grows, the correlations between the spatial and the temporal representations (both 1024-dimensional) can be captured more completely, thus the classification accuracy increases. It is a trade-off between compression and quality. But larger output dimension is not always good, since it makes the following fully connected layers hard to train. We observe that a 4096 output dimension is appropriate for both video datasets.

Fusion Method	Accuracy
Average	91.7%
Concatenation	92.4%
Element-wise Sum	92.3%
Compact Bilinear (d = 1024)	92.4%
Compact Bilinear (d = 2048)	92.9%
Compact Bilinear (d = 4096)	93.2%
Compact Bilinear (d = 8192)	93.2%

Table 3. Accuracy of various fusion methods on UCF101 (Split 1).

4.4. Temporal Fusion

Table 4 illustrates the impact of feeding temporal networks with longer sequences of optical flow data. Our model is implemented by making several copies of the individual network and combining them at the last convolutional layers. We have two observations here. First, among all these fusion strategies, spatiotemporal compact bilinear fusion presents the best performance. It is the first time that compact bilinear fusion is demonstrated effective for merging multi-path optical flow representations.

Second, these results explain why we design 3 subnetworks in out final architecture. The columns in Table 4 denotes the number of pathways before the fusion layer. Among all these models, a 3-path network with spatiotemporal compact bilinear fusion outperforms the others. We shall not cut down or increase the number of subnetworks. On one hand, the performance of the 3-path model is 2.3 points higher than that of the single-path. On the other hand, more subnetworks do not mean better results, since in this situation the spatial and temporal features may not correlate well with a very long sequence of optical flow data.

4.5. Spatiotemporal Attention

Attention pooling can effectively improve the overall classification accuracy by guiding the spatial network to

Fusion Method	1-path	3-path	5-path
Concatenation	87.0%	88.4%	88.5%
Element-wise Sum	-	87.9%	87.7%
Compact Bilinear	-	89.3%	89.2%

Table 4. A comparison of methods for merging multi-path temporal chunks. The columns represent the number of temporal chunks. All results are produced on UCF101 with optical flow data only.

attend to significant locations. Our experiments demonstrate that it can help avoid classification errors especially resulting from similar or chaotic backgrounds in static video frames. As shown in Table 5, our best implementation boosts the performance of the spatial pathway by 2.1 points.

Moreover, this set of experiments testify the value of compact bilinear fusion again. We initially intend to use temporal representations solely to generate attention weights. However, the result turns out to be a little lower than the original average pooling. We then try to merge temporal and spatial features in advance, while in this scenario the compact bilinear fusion performs surprisingly well.

Model	Spatial Accuracy
Average Pooling	84.5%
Att. Pooling (Temporal Only)	84.3%
Att. Pooling (Concatenation)	83.9%
Att. Pooling (Element-Wise Sum)	83.5%
Att. Pooling (Compact Bilinear)	86.6%

Table 5. The effect of applying attention pooling to the spatial network on UCF101. We feed the attention module with representations generated by various fusion methods.

4.6. Ablation Results

To testify the individual effect of fusion approaches we discuss above, we stack them one by one and test the overall performance. We set the baseline as the original two-stream CNNs that averages the outputs of the classifier layers. From Table 6, we observe that our spatiotemporal fusion method improves the average accuracy by 1.5 points. Furthermore, the proposed multi-path temporal fusion method results in another 0.4 points performance gain. At last, we apply spatiotemporal attention pooling and boost the final result to 94.2%. To sum up, all methods that we propose prove to be effective for video action recognition.

4.7. Final Results

Final results are obtained by following the testing scheme described in the standard two-stream method [22]. At first, 10 video snippets are randomly sampled and each of them contains 3 RGB images along with the corresponding 30 optical flow frames. We then enlarge the training

Model	A	B	C	D
ST Fusion	-	✓	✓	✓
Multi-T Fusion	-	-	✓	✓
Attention	-	-	-	✓
Accuracy	91.7%	93.2%	93.6%	94.2%

Table 6. Ablation results on UCF101 (Split 1). ST Fusion denotes two-stream spatiotemporal compact bilinear fusion. Multi-T Fusion denotes multi-path temporal fusion. Model A stands for the original two-stream CNNs, while the others stacks the proposed approaches one by one. In particular, D is the final architecture.

datasets by cropping the frames and flipping them to avoid over-fitting. All data belonging to one snippet is fed to the network to produce an estimate, and the video-level prediction is made by averaging over the 10 snippets.

Method	UCF101	HMDB51
Slow Fusion CNN [12]	65.4%	-
LRCN [5]	82.9%	-
C3D [28]	85.2%	-
Two-Stream (AlexNet) [22]	88.0%	59.4%
Two-Stream + LSTM [37]	88.6%	-
Two-Stream + Pooling [37]	88.2%	-
Transformation [33]	92.4%	62.0%
Two-Stream (VGG-16) [6]	90.6%	58.2%
Two-Stream + Fusion [6]	92.5%	65.4%
TSN (BN-Inception) [32]	94.0%	68.5%
Ours (VGG-16)	93.2%	66.1%
Ours (ResNet-50)	93.8%	66.5%
Ours (BN-Inception)	**94.6%**	**68.9%**

Table 7. Performance comparison with the state-of-the-art.

We compare the performance of our final architecture with the state-of-the-art in Table 7. Our best implementation based on BN-Inception improves the average accuracy by 0.6% on UCF101 and 0.4% on HMDB51. Someone may cast doubt on it and own this performance boost to the very deep models. To testify that our method is generally effective, we additionally use the same base architecture (VGG-16) as the previous two-stream CNNs architectures. Both based on VGG-16, our result (**93.2%**) is still competitive to the original two-stream fusion [6] (**92.5%**). Again, both based on BN-Inception, our new result (**94.6%**) is superior to the state-of-the-art [32] (**94.0%**). This result also illustrate that our approaches are not any deep-network exclusive, but can be widely applied to many fancy CNN models.

Some representative examples of the classification results are shown in Figure 4. The first two subplots demonstrate the effectiveness of the spatiotemporal attention module. As mentioned above, the original two-stream network is easily fooled by common background. For instance, it regards FrontCrawl as BreastStroke, since the swimming pool

appears to be a dominate feature. But in our model, these spatially ambiguous classes can be separated by exploiting motion information to extract attended regions of the activities. The last two subplots illustrate another strength of our pyramid network. Thanks to the multi-path temporal fusion, it produces more global features over longer video sequences and can easily differentiate actions that look similar in short-term snippets but may vary substantially in a long-term. Additionally, Figure 3 gives typical examples of categories that are difficult to classify. In the first case, the network sometimes regards Skiing as SkateBoarding. We can observe that the main difference of these two classes resides in the scene environment. A possible reason is that CNN is not robust to the color bias of the image background. There is no evidence that the attention truly ignores the background and harms the classification performance. On the contrary, it offers the fusion pyramid some useful and additional cues for accurate predictions. If observing Figure 3 carefully, one can find the confidence of the correct category (Skiing) has actually increased. This shows that one component may amend the error of others in the fusion pyramid. Moreover, the second example indicates that some categories, like PizzaTossing and Nunchucks, can only be disambiguated by taking advantage of a fine-grained recognition. That is to say, detecting the detailed objects in connection with the on-going actions is also important. The study towards this issue may reside in our future research.

Furthermore, we adopt t-SNE techniques to visualize feature vectors that are learned at different levels of the spatiotemporal pyramid, as shown in Figure 5. We observe that the spatiotemporal attention can indeed improve the feature quality of the spatial stream, and the spatiotemporal com-
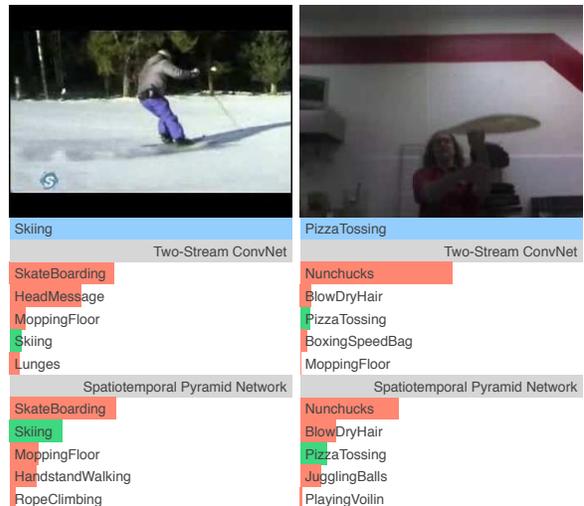
Figure 3. Examples of categories that are difficult to classify. Even for misclassification cases, the confidence of the correct category (green) has increased by our method.

BlowDryHair	FrontCrawl	BoxingPunchingBag	PullUps
Two-Stream ConvNet	Two-Stream ConvNet	Two-Stream ConvNet	Two-Stream ConvNet
Haircut	BreastStroke	BoxingSpeedBag	RopeClimbing
BlowDryHair	FrontCrawl	BoxingPunchingBag	PullUps
ShavingBeard	Kayaking	BlowDryHair	RockClimbingIndoor
ApplyEyeMakeup	CliffDiving	JugglingBalls	BoxingSpeedBag
RopeClimbing	Diving	Archery	BoxingPunchingBag
Spatiotemporal Pyramid Network	Spatiotemporal Pyramid Network	Spatiotemporal Pyramid Network	Spatiotemporal Pyramid Network
BlowDryHair	FrontCrawl	BoxingPunchingBag	PullUps
Haircut	BreastStroke	BoxingSpeedBag	RopeClimbing
ShavingBeard	Kayaking	BlowDryHair	HandstandPushups
PlayingGuitar	CliffDiving	Nunchucks	WallPushups
HeadMessage	Diving	JugglingBalls	RockClimbingIndoor

Figure 4. A comparison of top-5 predictions between the baseline and our pyramid network on UCF-101. The blue bars denote the ground truth labels, the green bars indicate correct classifications and the red stand for incorrect cases. The length of each bar shows its confidence. With pyramid network, some errors can be eliminated by taking advantage of image attention (right two), while some other categories can be disambiguated by fusing long-term temporal features (left two).

Spatial   Spatiotemporal Attention   Spatiotemporal Pyramid Fusion
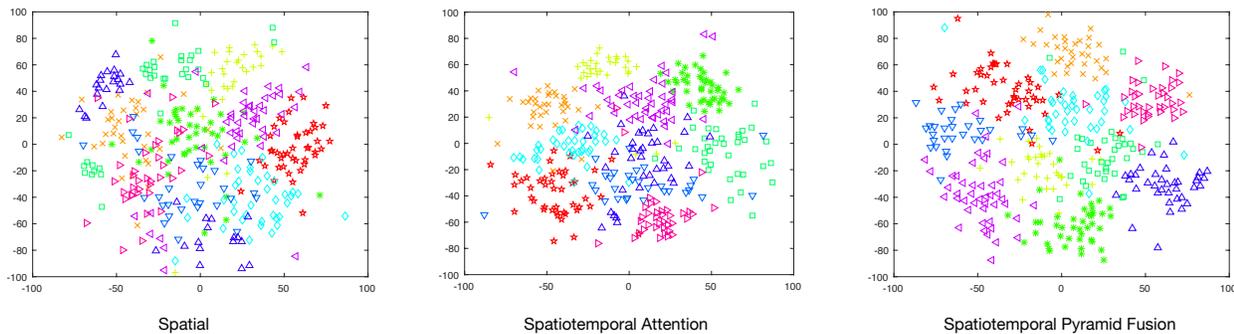
Figure 5. The t-SNE results of 10 classes randomly selected from UCF101. The left figure shows representations of an individual spatial network; the middle figure is obtained by adding the spatiotemporal attention method on it; and the right one denotes the results of our final spatiotemporal pyramid architecture. We use features of the classifier layer for all these cases.

pact bilinear fusion at the top of the pyramid can further increase the discriminative performance.

5. Conclusions

We propose a spatiotemporal pyramid network to combine the spatial and temporal features and make them reinforce each other. From the architecture perspective, our network is hierarchical, consisting of multiple fusion strategies at different abstraction levels. These fusion modules are trained as a whole to maximally complementing each other. A series of ablation studies validate the importance of each fusion technique. From the technical perspective, we introduce the spatiotemporal compact bilinear operator into video analysis tasks. This operator can learn element-wise interactions between the spatial and temporal features. We extensively show its benefit over other fusion methods, such as concatenation and element-wise sum. Our spatiotemporal pyramid network achieves the state-of-the-art performance on UCF101 and HMDB51.

References

[1] M. Charikar, K. Chen, and M. Farach-Colton. *Finding Frequent Items in Data Streams*, pages 693–703. Springer Berlin Heidelberg, 2002.

[2] G. Chéron, I. Laptev, and C. Schmid. P-cnn: Pose-based cnn features for action recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3218–3226, 2015.

[3] M. Cimpoi, S. Maji, and A. Vedaldi. Deep filter banks for texture recognition and segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3828–3836, 2015.

[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.

[5] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2625–2634, 2015.

[6] C. Feichtenhofer, A. Pinz, and A. Zisserman. Convolutional two-stream network fusion for video action recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[7] Y. Gao, O. Beijbom, N. Zhang, and T. Darrell. Compact bilinear pooling. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[8] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[9] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2015.

[10] S. Ji, W. Xu, M. Yang, and K. Yu. 3d convolutional neural networks for human action recognition. *IEEE transactions on pattern analysis and machine intelligence*, 35(1):221–231, 2013.

[11] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 675–678. ACM, 2014.

[12] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale video classification with convolutional neural networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 1725–1732, 2014.

[13] 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.

[14] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre. Hmdb: a large video database for human motion recognition. In *2011 International Conference on Computer Vision*, pages 2556–2563. IEEE, 2011.

[15] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.

[16] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[17] T.-Y. Lin, A. RoyChowdhury, and S. Maji. Bilinear cnn models for fine-grained visual recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1449–1457, 2015.

[18] J. Lu, J. Yang, D. Batra, and D. Parikh. Hierarchical question-image co-attention for visual question answering. In *Advances in Neural Information Processing Systems (NIPS)*, 2016.

[19] F. Perronnin, J. Sánchez, and T. Mensink. Improving the fisher kernel for large-scale image classification. In *European conference on computer vision*, pages 143–156. Springer, 2010.

[20] N. Pham and R. Pagh. Fast and scalable polynomial kernels via explicit feature maps. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '13, pages 239–247, New York, NY, USA, 2013. ACM.

[21] S. Sharma, R. Kiros, and R. Salakhutdinov. Action recognition using visual attention. In *ICLR*, 2016.

[22] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in Neural Information Processing Systems*, pages 568–576, 2014.

[23] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2015.

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

[25] N. Srivastava, E. Mansimov, and R. Salakhutdinov. Unsupervised learning of video representations using lstms. *CoRR, abs/1502.04681*, 2, 2015.

[26] L. Sun, K. Jia, D.-Y. Yeung, and B. E. Shi. Human action recognition using factorized spatio-temporal convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4597–4605, 2015.

[27] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9, 2015.

[28] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3d convolutional networks. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 4489–4497. IEEE, 2015.

[29] S. Venugopalan, M. Rohrbach, J. Donahue, R. Mooney, T. Darrell, and K. Saenko. Sequence to sequence-video to

text. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4534–4542, 2015.

[30] H. Wang and C. Schmid. Action recognition with improved trajectories. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3551–3558, 2013.

[31] L. Wang, Y. Qiao, and X. Tang. Action recognition with trajectory-pooled deep-convolutional descriptors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4305–4314, 2015.

[32] L. Wang, Y. Xiong, and et al. Temporal segment networks: Towards good practices for deep action recognition. In *ECCV*, 2016.

[33] X. Wang, A. Farhadi, and A. Gupta. Actions transformations. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 2658–2667, 2016.

[34] P. Weinzaepfel, Z. Harchaoui, and C. Schmid. Learning to track for spatio-temporal action localization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3164–3172, 2015.

[35] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhutdinov, R. S. Zemel, and Y. Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2015.

[36] Z. Yang, X. He, J. Gao, L. Deng, and A. Smola. Stacked attention networks for image question answering. *arXiv preprint arXiv:1511.02274*, 2015.

[37] J. Yue-Hei Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and G. Toderici. Beyond short snippets: Deep networks for video classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4694–4702, 2015.

[38] B. Zhang, L. Wang, Z. Wang, Y. Qiao, and H. Wang. Real-time action recognition with enhanced motion vector cnns. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.