Learning Spatio-Temporal Representation with Pseudo-3D Residual Networks *

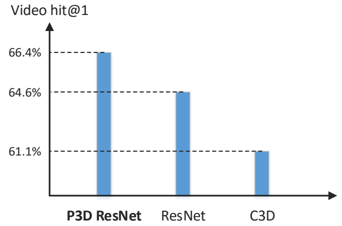
Zhaofan Qiu †, Ting Yao ‡, and Tao Mei ‡
† University of Science and Technology of China, Hefei, China
‡ Microsoft Research, Beijing, China
zhaofanqiu@gmail.com, {tiyao, tmei}@microsoft.com

Abstract

Convolutional Neural Networks (CNN) have been regarded as a powerful class of models for image recognition problems. Nevertheless, it is not trivial when utilizing a CNN for learning spatio-temporal video representation. A few studies have shown that performing 3D convolutions is a rewarding approach to capture both spatial and temporal dimensions in videos. However, the development of a very deep 3D CNN from scratch results in expensive computational cost and memory demand. A valid question is why not recycle off-the-shelf 2D networks for a 3D CNN. In this paper, we devise multiple variants of bottleneck building blocks in a residual learning framework by simulating $3 \times 3 \times 3$ convolutions with $1 \times 3 \times 3$ convolutional filters on spatial domain (equivalent to 2D CNN) plus $3 \times 1 \times 1$ convolutions to construct temporal connections on adjacent feature maps in time. Furthermore, we propose a new architecture, named Pseudo-3D Residual Net (P3D ResNet), that exploits all the variants of blocks but composes each in different placement of ResNet, following the philosophy that enhancing structural diversity with going deep could improve the power of neural networks. Our P3D ResNet achieves clear improvements on Sports-1M video classification dataset against 3D CNN and frame-based 2D CNN by 5.3% and 1.8%, respectively. We further examine the generalization performance of video representation produced by our pre-trained P3D ResNet on five different benchmarks and three different tasks, demonstrating superior performances over several state-of-the-art techniques.

1. Introduction

Today's digital contents are inherently multimedia: text, audio, image, video and so on. Images and videos, in particular, become a new way of communication between Internet users with the proliferation of sensor-rich mobile devices. This has encouraged the development of advanced techniques for a broad range of multimedia understanding applications. A fundamental progress that underlies the success of these technological advances is representation learning. Recently, the rise of Convolutional Neural Networks (CNN) convincingly demonstrates high capability of learning visual representation especially in image domain. For instance, an ensemble of residual nets [7] achieves 3.57% top-5 error on the ImageNet test set, which is even lower than 5.1% of the reported human-level performance. Nevertheless, video is a temporal sequence of frames with large variations and complexities, resulting in difficulty in learning a powerful and generic spatio-temporal representation.



Method	Depth	Model size
C3D	11	321MB
ResNet	152	235MB
P3D ResNet	199	261MB

Figure 1. Comparisons of different models on Sports-1M dataset in terms of accuracy, model size and the number of layers.

One natural way to encode spatio-temporal information in videos is to extend the convolution kernels in CNN from 2D to 3D and train a brand new 3D CNN. As such, the networks have access not only the visual appearance present in each video frame, but also the temporal evolution across consecutive frames. While encouraging performances are reported in recent studies [8, 31, 33], the training of 3D CNN is very computationally expensive and the model size also has a quadratic growth compared to 2D CNN. Take a widely adopted 11-layer 3D CNN, i.e., C3D [31] networks, as an example, the model size reaches 321MB which is even larger than that (235MB) of a 152-layer 2D ResNet (ResNet-152) [7], making it extremely difficult to train a very deep 3D CNN. More importantly, directly fine-tuning ResNet-152 with frames in Sports-1M dataset [10] may achieve better accuracy than C3D trained on videos from scratch as shown in Figure 1. Another alternative solution of producing spatio-temporal video representation is to utilize pool-

*This work was performed when Zhaofan Qiu was visiting Microsoft Research as a research intern.

ing strategy or Recurrent Neural Networks (RNN) over the representations of frames, which are often the activations of a 2D CNN's last pooling layer or fully-connected layers. This category of approaches, however, only build temporal connections on the high-level features at the top layer while leaving the correlations in the low-level forms, e.g., corners or edges at the bottom layers, not fully exploited.

We demonstrate in this paper that the above limitations can be mitigated by devising a family of bottleneck building blocks that leverages both spatial and temporal convolutional filters. Specifically, the key component in each block is a combination of one $1 \times 3 \times 3$ convolutional layer and one layer of $3 \times 1 \times 1$ convolutions in a parallel or cascaded fashion, that takes the place of a standard $3 \times 3 \times 3$ convolutional layer. As such, the model size is significantly reduced and the advantages of pre-learnt 2D CNN in image domain could also be fully leveraged by initializing the $1 \times 3 \times 3$ convolutional filters with 3×3 convolutions in 2D CNN. Furthermore, we propose a novel Pseudo-3D Residual Net (P3D ResNet) that composes each designed block in different placement throughout a whole ResNet-like architecture to enhance the structural diversity of the network. As a result, the temporal connections in our P3D ResNet are constructed at every level from bottom to top and the learnt video representations encapsulate information related to objects, scenes and actions in videos, making them generic for various video analysis tasks.

The main contribution of this work is the proposal of a family of bottleneck building blocks that simulates 3D convolutions in an economic and effective way. This also leads to the elegant view of how different blocks should be placed for learning very deep networks and a new P3D ResNet is presented for video representation learning. Through an extensive set of experiments, we demonstrate that our P3D ResNet outperforms several state-of-the-art models on five different benchmarks and three different tasks.

2. Related Work

We briefly group the methods for video representation learning into two categories: hand-crafted and deep learning-based methods.

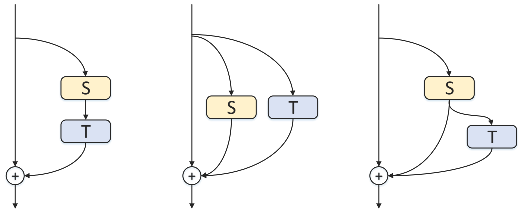
Hand-crafted representation learning methods usually start by detecting spatio-temporal interest points and then describe these points with local representations. In this scheme, Space-Time Interest Points (STIP) [15], Histogram of Gradient and Histogram of Optical Flow [16], 3D Histogram of Gradient [11] and SIFT-3D [23] are proposed by extending representations from image domain to measure the temporal dimension of 3D volumes. Recently, Wang *et al.* propose dense trajectory features, which densely sample local patches from each frame at different scales and then track them in a dense optical flow field [34].

The most recent approaches for video representation learning are to devise deep architectures. Karparthy *et al.* stack CNN-based frame-level representations in a fixed size of windows and then leverage spatio-temporal convolutions for learning video representation [10]. In [25], the famous two-stream architecture is devised by applying two CNN architectures separately on visual frames and staked optical flows. This architecture is further extended by exploiting multi-granular structure [17, 18, 21], convolutional fusion [6], key-volume mining [39] and temporal segment networks [36] for video representation learning. In the work by Wang et al. [35], the local ConvNet responses over the spatio-temporal tubes centered at the trajectories are pooled as the video descriptors. Fisher Vector [20] is then used to encode these local descriptors to a global video representation. Recently, the LSTM-RNN networks have been successfully employed for modeling temporal dynamics in videos. In [9, 37], temporal pooling and stacked LSTM network are leveraged to combine frame-level (optical flow images) representation and discover long-term temporal relationships for learning a more robust video representation. Srivastava *et al.* [28] further formulate the video representation learning task as an autoencoder model based on the encoder and decoder LSTMs.

It can be observed that most aforementioned deep learning-based methods treat video as a frame/optical flow image sequence for video representation learning while leaving the temporal evolution across consecutive frames not fully exploited. To tackle this problem, 3D CNN proposed by Ji *et al.* [8] is one of the earlier works to directly learn the spatio-temporal representation of a short video clip. Later in [31], Tran *et al.* devise a widely adopted 11-layer 3D CNN (C3D) for learning video representation over 16-frame video clips in the context of large-scale supervised video datasets, and temporal convolutions across longer clips (100 frames) are further exploited in [33]. However, the capacity of existing 3D CNN architectures is extremely limited with expensive computational cost and memory demand, making it hard to train a very deep 3D CNN. Our method is different that we not only propose the idea of simulating 3D convolutions with 2D spatial convolutions plus 1D temporal connections which is more economic, but also integrate this design into a deep residual learning framework for video representation learning.

3. P3D Blocks and P3D ResNet

In this section we firstly define the 3D convolutions for video representation learning which can be naturally decoupled into 2D spatial convolutions to encode spatial information and 1D temporal convolutional filters for temporal dimension. Then, a new family of bottleneck building blocks, namely Pseudo-3D (P3D), to leverage both spatial and temporal convolutional filters is devised in the residual learning framework. Finally, we develop a novel Pseudo-3D Residu-



(a) P3D-A (b) P3D-B (c) P3D-C

Figure 2. Three designs of Pseudo-3D blocks.

al Net (P3D ResNet) composing each P3D block at different placement in ResNet-like architecture and further compare its several variants through experimental studies in terms of both performance and time efficiency.

3.1. 3D Convolutions

Given a video clip with the size of $c \times l \times h \times w$ where c, l, h and w denotes the number of channels, clip length, height and width of each frame, respectively, the most natural way to encode the spatio-temporal information is to utilize 3D convolutions [8, 31]. 3D convolutions simultaneously model the spatial information like 2D filters and construct temporal connections across frames. For simplicity, we denote the size of 3D convolutional filters as $d \times k \times k$ where d is the temporal depth of kernel and k is the kernel spatial size. Hence, suppose we have 3D convolutional filters with size of $3 \times 3 \times 3$, it can be naturally decoupled into $1 \times 3 \times 3$ convolutional filters equivalent to 2D CNN on spatial domain and $3 \times 1 \times 1$ convolutional filters like 1D CNN tailored to temporal domain. Such decoupled 3D convolutions can be regarded as a Pseudo 3D CNN, which not only reduces the model size significantly, but also enables the pre-training of 2D CNN from image data, endowing Pseudo 3D CNN more power of leveraging the knowledge of scenes and objects learnt from images.

3.2. Pseudo-3D Blocks

Inspired by the recent successes of Residual Networks (ResNet) [7] in numerous challenging image recognition tasks, we develop a new family of building modules named Pseudo-3D (P3D) blocks to replace 2D Residual Units in ResNet, pursuing spatio-temporal encoding in ResNet-like architectures for videos. Next, we will recall the basic design of Residual Units in ResNet, followed by presenting how to devise our P3D blocks. The bottleneck building architecture on each P3D block is finally elaborated.

Residual Units. ResNet consists of many staked Residual Units and each Residual Unit could be generally given by

$$\mathbf{x}_{t+1} = \mathbf{h}\left(\mathbf{x}_t\right) + \mathbf{F}\left(\mathbf{x}_t\right), \tag{1}$$

where $\mathbf{x}_t$ and $\mathbf{x}_{t+1}$ denote the input and output of the t-th Residual Unit, $\mathbf{h}\left(\mathbf{x}_t\right) = \mathbf{x}_t$ is an identity mapping and $\mathbf{F}$ is a non-linear residual function. Hence, Eq.(1) can be rewritten as

$$\left(\mathbf{I} + \mathbf{F}\right) \cdot \mathbf{x}_t = \mathbf{x}_t + \mathbf{F} \cdot \mathbf{x}_t := \mathbf{x}_t + \mathbf{F}\left(\mathbf{x}_t\right) = \mathbf{x}_{t+1}, \tag{2}$$

where $\mathbf{F} \cdot \mathbf{x}_t$ represents the result of performing residual function $\mathbf{F}$ over $\mathbf{x}_t$. The main idea of ResNet is to learn the additive residual function $\mathbf{F}$ with reference to the unit inputs $\mathbf{x}_t$ which is realized through a shortcut connection, instead of directly learning unreferenced non-linear functions.

P3D Blocks design. To develop each 2D Residual Unit in ResNet into 3D architectures for encoding spatio-temporal video information, we modify the basic Residual Unit in ResNet following the principle of Pseudo 3D as introduced in Section 3.1 and devise several Pseudo-3D Blocks. The modification is not straightforward for involvement of two design issues. The first issue is about whether the modules of 2D filters on spatial dimension ($\mathbf{S}$) and 1D filters on temporal domain ($\mathbf{T}$) should directly or indirectly influence each other. Direct influence within the two types of filters means that the output of spatial 2D filters is connected as the input to the temporal 1D filters (i.e., in a cascaded manner). Indirect influence between the two filters decouples the connection such that each kind of filters is on a different path of the network (i.e., in a parallel fashion). The second issue is whether the two kinds of filters should both directly influence the final output. As such, direct influence in this context denotes that the output of each type of filters should be directly connected to the final output.

Based on the two design issues, we derive three different P3D blocks as depicted in Figure 2, respectively, named as P3D-A to P3D-C. Detailed comparisons about their architectures are provided as following:

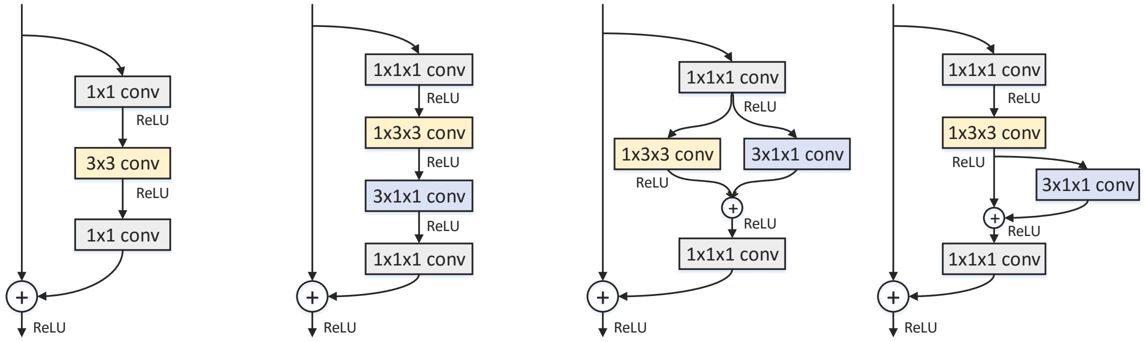
(1) P3D-A: The first design considers stacked architecture by making temporal 1D filters ($\mathbf{T}$) follow spatial 2D filters ($\mathbf{S}$) in a cascaded manner. Hence, the two kinds of filters can directly influence each other in the same path and only the temporal 1D filters are directly connected to the final output, which could be generally given by

$$\left(\mathbf{I} + \mathbf{T} \cdot \mathbf{S}\right) \cdot \mathbf{x}_t := \mathbf{x}_t + \mathbf{T}\left(\mathbf{S}\left(\mathbf{x}_t\right)\right) = \mathbf{x}_{t+1}. \tag{3}$$

(2) P3D-B: The second design is similar to the first one except that indirect influence between two filters are adopted and both filters are at different pathways in a parallel fashion. Although there is no direct influence between $\mathbf{S}$ and $\mathbf{T}$, both of them are directly accumulated into the final output, which could be expressed as

$$\left(\mathbf{I} + \mathbf{S} + \mathbf{T}\right) \cdot \mathbf{x}_t := \mathbf{x}_t + \mathbf{S}\left(\mathbf{x}_t\right) + \mathbf{T}\left(\mathbf{x}_t\right) = \mathbf{x}_{t+1}. \tag{4}$$

(3) P3D-C: The last design is a compromise between P3D-A and P3D-B, by simultaneously building the direct influences among $\mathbf{S}$, $\mathbf{T}$ and the final output. Specifically, to enable the direct connection between $\mathbf{S}$ and final output based on the cascaded P3D-A architecture, we establish a



(a) Residual Unit [7] (b) P3D-A (c) P3D-B (d) P3D-C

Figure 3. Bottleneck building blocks of Residual Unit and our Pseudo-3D.

shortcut connection from $\mathbf{S}$ to the final output, making the output $\mathbf{x}_{t+1}$ as

$$(\mathbf{I}+\mathbf{S}+\mathbf{T}\cdot\mathbf{S})\cdot\mathbf{x}_t := \mathbf{x}_t+\mathbf{S}\left(\mathbf{x}_t\right)+\mathbf{T}\left(\mathbf{S}\left(\mathbf{x}_t\right)\right)=\mathbf{x}_{t+1}. \quad (5)$$

Bottleneck architectures. When specifying the architecture of 2D Residual Unit, the basic 2D block is modified with a bottleneck design for reducing the computation complexity. In particular, as shown in Figure 3(a), instead of a single spatial 2D filters (3 × 3 convolutions), the Residual Unit adopts a stack of 3 layers including 1 × 1, 3 × 3, and 1×1 convolutions, where the first and last 1×1 convolutional layers are applied for reducing and restoring dimensions of input sample, respectively. Such bottleneck design makes the middle 3 × 3 convolutions as a bottleneck with smaller input and output dimensions. Thus, we follow this elegant recipe and utilize the bottleneck design to implement our proposed P3D blocks. Similar in spirit, for each P3D block which purely consists of one spatial 2D filters (1 × 3 × 3 convolutions) and one temporal 1D filters (3 × 1 × 1 convolutions), we additionally place two 1 × 1 × 1 convolutions at both ends of the path, which are responsible for reducing and then increasing the dimensions. Accordingly, the dimensions of the input and output of both the spatial 2D and temporal 1D filters are reduced with this bottleneck design. The detailed bottleneck building architectures on all the three P3D blocks are illustrated in Figure 3(b) to 3(d).

3.3. Pseudo-3D ResNet

In order to verify the merit of the three P3D blocks, we first develop three P3D ResNet variants, i.e., P3D-A ResNet, P3D-B ResNet and P3D-C ResNet by replacing all the Residual Units in a 50-layer ResNet (ResNet-50) [7] with one certain kind of P3D block, respectively. The comparisons of performance and time efficiency between the basic ResNet-50 and the three P3D ResNet variants are presented. Then, a complete version of P3D ResNet is proposed by mixing all the three P3D blocks from the viewpoint of structural diversity.

Table 1. Comparisons of ResNet-50 and different Pseudo-3D ResNet variants in terms of model size, speed, and accuracy on UCF101 (split1). The speed is reported on one NVidia K40 GPU.

Method	**Model size**	**Speed**	**Accuracy**
ResNet-50	92MB	15.0 frame/s	80.8%
P3D-A ResNet	98MB	9.0 clip/s	83.7%
P3D-B ResNet	98MB	8.8 clip/s	82.8%
P3D-C ResNet	98MB	8.6 clip/s	83.0%
P3D ResNet	98MB	8.8 clip/s	84.2%

Comparisons between P3D ResNet variants. The comparisons are conducted on UCF101 [27] video action recognition dataset. Specifically, the architecture of ResNet-50 is fine-tuned on UCF101 video data. We set the input as 224 × 224 image which is randomly cropped from the resized 240 × 320 video frame. Moreover, following [36], we freeze the parameters of all Batch Normalization layers except for the first one and add an extra dropout layer with 0.9 dropout rate to reduce the effect of over-fitting. After fine-tuning ResNet-50, the networks will predict one score for each frame and the video-level prediction score is calculated by averaging all frame-level scores. The architectures of three P3D ResNet variants are all initialized with ResNet-50 except for the additional temporal convolutions and are further fine-tuned on UCF101. For each P3D ResNet variant, the dimension of input video clip is set as 16 × 160 × 160 which is randomly cropped from the resized non-overlapped 16-frame clip with the size of 16 × 182 × 242. Each frame/clip is randomly horizontally flipped for data augmentation. In the training stage, we set each mini-batch as 128 frames/clips, which are implemented with multiple GPUs in parallel. The network parameters are optimized by standard SGD and the initial learning rate is set as 0.001, which is divided by 10 after every 3K iterations. The training is stopped after 7.5K iterations.

Table 1 shows the performance and time efficiency of ResNet-50 and our Pseudo-3D ResNet variants on UCF101. Overall, all the three P3D ResNet variants (i.e., P3D-A ResNet, P3D-B ResNet and P3D-C ResNet) exhibit better performance than ResNet-50 with only a small increase in

Table 2. Comparisons in terms of pre-train data, clip length, Top-1 clip-level accuracy and Top-1&5 video-level accuracy on Sports-1M.

Method	Pre-train Data	Clip Length	Clip hit@1	Video hit@1	Video hit@5
Deep Video (Single Frame) [10]	ImageNet1K	1	41.1%	59.3%	77.7%
Deep Video (Slow Fusion) [10]	ImageNet1K	10	41.9%	60.9%	80.2%
Convolutional Pooling [37]	ImageNet1K	120	70.8%	72.3%	90.8%
C3D [31]	–	16	44.9%	60.0%	84.4%
C3D [31]	I380K	16	46.1%	61.1%	85.2%
ResNet-152 [7]	ImageNet1K	1	46.5%	64.6%	86.4%
P3D ResNet (ours)	ImageNet1K	16	47.9%	66.4%	87.4%

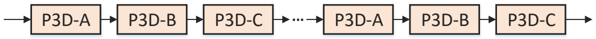


Figure 4. P3D ResNet by interleaving P3D-A, P3D-B and P3D-C.

model size. The results basically indicate the advantage of exploring spatio-temporal information by our P3D blocks. Moreover, the speed of our P3D ResNet variants is very fast and could reach 8.6 $\sim$ 9.0 clips per second.

Mixing different P3D Blocks. Further inspired from the recent success of pursuing structural diversity in the design of very deep networks [38], we devise a complete version of P3D ResNet by mixing different P3D blocks in the architecture to enhance structural diversity, as depicted in Figure 4. Particularly, we replace Residual Units with a chain of our P3D blocks in the order P3D-A$\rightarrow$P3D-B$\rightarrow$P3D-C. Table 1 also details the performance and speed of the complete P3D ResNet. By additionally pursuing structural diversity, P3D ResNet makes the absolute improvement over P3D-A ResNet, P3D-B ResNet and P3D-C ResNet by 0.5%, 1.4% and 1.2% in accuracy respectively, indicating that enhancing structural diversity with going deep could improve the power of neural networks.

4. Spatio-Temporal Representation Learning

We further validate the complete design of our P3D ResNet on a deeper 152-layer ResNet [7] and then produce a generic spatio-temporal video representation. The learning of P3D ResNet here was conducted on Sports-1M dataset [10], which is one of the largest video classification benchmark. It roughly contains about 1.13 million videos annotated with 487 Sports labels. There are 1K-3K videos per label and approximately 5% of the videos are with more than one label. Please also note that about 9.2% video URLs were dead when we downloaded the videos. Hence, we conducted the experiments on the remaining 1.02 million videos and followed the official split, i.e., 70%, 10% and 20% for training, validation and test set, respectively.

Network Training. For efficient training on the large Sports-1M training set, we randomly select five 5-second short videos from each video in the set. During training, the settings of data augmentation and mini-batch are the same as those in Section 3.3 except that the dropout rate is set as 0.1. The learning rate is also initialized as 0.001, and divided by 10 after every 60K iterations. The optimization will be complete after 150K batches.

Network Testing. We evaluate the performance of the learnt P3D ResNet by measuring video/clip classification accuracy on the test set. Specifically, we randomly sample 20 clips from each video and adopt a single center crop per clip, which is propagated through the network to obtain a clip-level prediction score. The video-level score is computed by averaging all the clip-level scores of a video.

We compare the following approaches for performance evaluation: (1) Deep Video (Single Frame) and (Slow Fusion) [10]. The former performs a CNN which is similar to the architecture in [14] on one single frame from each clip to predict a clip-level score and fuses multiple frames in each clip with different temporal extent throughout the network to achieve the clip-level prediction. (2) Convolutional Pooling [37] exploits max-pooling over the final convolutional layer of GoogleNet [30] across each clip's frames. (3) C3D [31] utilizes 3D convolutions on a clip volume to model the temporal information and the whole architecture could be trained on Sports-1M dataset from scratch or fine-tuned from the pre-trained model on I380K internal dataset collected in [31]. (4) ResNet-152 [7]. In this run, a 152-layer ResNet is fine-tuned and employed on one frame from each clip to produce a clip-level score.

The performances and comparisons are summarized in Table 2. Overall, our P3D ResNet leads to a performance boost against ResNet-152 (2D CNN) and C3D (3D CNN) by 1.8% and 5.3% in terms of top-1 video-level accuracy, respectively. The results basically indicate the advantage of exploring spatio-temporal information by decomposing 3D learning into 2D convolutions in spatial space and 1D operations in temporal dimension. As expected, Deep Video (Slow Fusion) fusing temporal information throughout the networks exhibits better performance than Deep Video (Single Frame) which exploits only one single frame. Though the three runs of Deep Video (Slow Fusion), Convolutional Pooling and our P3D ResNet all capitalizes on temporal fusion, they are fundamentally different in the way of performing temporal connections. The performance of Deep Video (Slow Fusion) is as a result of carrying out temporal convolutions on spatial convolutions to compute activations, while Convolutional Pooling is by simply max-pooling the outputs of final convolutional layer across temporal frames. As indicated by the results, our P3D ResNet employing different combinations of spatial and temporal

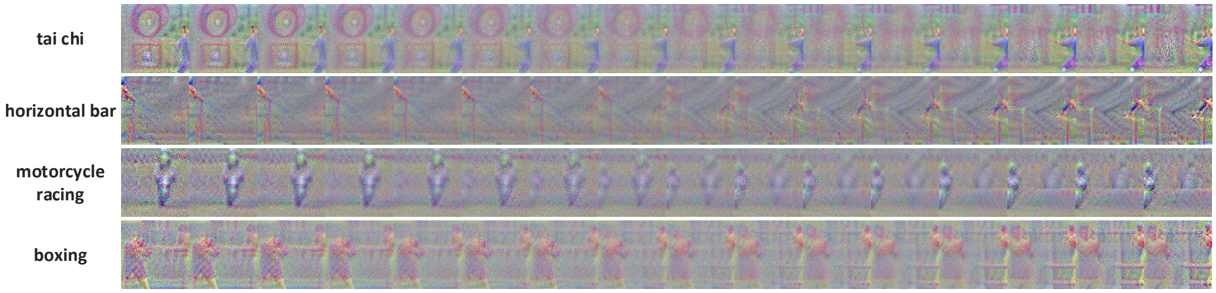


Figure 5. Visualization of class knowledge inside P3D ResNet model by using DeepDraw [1]. Four categories, i.e., tai chi, horizontal bar, motorcycle racing and boxing, are selected for visualization.

convolutions improves Deep Video (Slow Fusion). This somewhat indicates that P3D ResNet is benefited from the principle of structural diversity in network design. It is also not surprise that the performances of P3D ResNet are still lower than Convolutional Pooling which performs temporal pooling on 120 frames' clips with frame rate of 1 fps, making the clip length over 120s. In contrast, we take 16 consecutive frames as a basic unit which only covers less than 0.5s but has strong spatio-temporal connections, making our P3D ResNet with better generalization capability.

Figure 5 further visualizes the insights in the learnt P3D ResNet model. Following [36], we adopt DeepDraw toolbox [1], which conducts iterative gradient ascent on the input clip of white noises. During learning, it evaluates the model's violation of class label and back-propagates the gradients to modify the input clip. Thus, the final generated input clip could be regarded as the visualization of class knowledge inside P3D ResNet. We select four categories, i.e., tai chi, horizontal bar, motorcycle racing and boxing, for visualization. As illustrated in the figure, P3D ResNet model could capture both spatial visual patterns and temporal motion. Take the category of tai chi as an example, our model generates a video clip in which a person is displaying different poses, depicting the process of this action.

P3D ResNet Representation. After training our P3D ResNet architecture on Sports-1M dataset, the networks could be utilized as a generic representation extractor for any video analysis tasks. Given a video, we select 20 video clips and each clip is with 16-frame long. Each video clip is then input into the learnt P3D ResNet architecture and the 2,048 dimensional activations of pool5 layer are output as the representation of this clip. Finally, all the clip-level representations in a video are averaged to produce a 2,048 dimensional video representation. We refer to this representation as P3D ResNet representation in the following evaluations unless otherwise stated.

5. Video Representation Evaluation

Next, we evaluate our P3D ResNet video representation on three different tasks and five popular datasets, i.e., UCF101 [27], ActivityNet [2], ASLAN [13], YUPENN [3] and Dynamic Scene [24]. UCF101 and ActivityNet are two of the most popular video action recognition benchmarks. UCF101 consists of 13,320 videos from 101 action categories. Three training/test splits are provided by the dataset organisers and each split in UCF101 includes about 9.5K training and 3.7K test videos. The ActivityNet dataset is a large-scale video benchmark for human activity understanding. The latest released version of the dataset (v1.3) is exploited, which contains 19,994 videos from 200 activity categories. The 19,994 videos are divided into 10,024, 4,926 and 5,044 videos for training, validation and test set, respectively. It is also worth noting that the labels of test set are not publicly available and thus the performances on ActivityNet dataset are all reported on validation set.

ASLAN is a dataset on action similarity labeling task, which is to predict the similarity between videos. The dataset includes 3,697 videos from 432 action categories. We follow the strategy of 10-fold cross validation with the official data splits on this set. Furthermore, YUPENN and Dynamic Scene are two sets for the scenario of scene recognition. In between, YUPENN is comprised of 14 scene categories each containing 30 videos. Dynamic Scene consists of 13 scene classes with 10 videos per class. The training and test procedures on both datasets follow the standard leave-one-video-out protocol.

Comparison with the state-of-the-art. We first compare with several state-of-the-art techniques in the context of video action recognition on three splits of UCF101 and ActivityNet validation set. The performance comparisons are summarized in Table 3 and 4, respectively. We briefly group the approaches on UCF101 into three categories: end-to-end CNN architectures which are fine-tuned on UCF101 in the upper rows, CNN-based video representation extractors with linear SVM classifier in the middle rows and approaches fused with IDT in the bottom rows. It is worth noting that most recent end-to-end CNN architectures on UCF101 often employ and fuse two or multiple types of inputs, e.g., frame, optical flow or even audio. Hence, the performances by exploiting only video frames and late fusing the scores on two inputs of video frames plus optical flow are both reported. As shown in Table 3,

Table 3. Performance comparisons with the state-of-the-art methods on UCF101 (3 splits). TSN: Temporal Segment Networks [36]; TDD: Trajectory-pooled Deep-convolutional Descriptor [35]; IDT: Improved Dense Trajectory [34]. We group the approaches into three categories, i.e., end-to-end CNN architectures which are fine-tuned on UCF101 at the top, CNN-based video representation extractors with linear SVM classifier in the middle and approaches fused with IDT at the bottom. For the methods in the first direction, we report the performance of only taking frames and frames plus optical flow (in brackets) as inputs, respectively.

Method	**Accuracy**
End-to-end CNN architecture with fine-tuning	
Two-stream ConvNet [25]	73.0% (88.0%)
Factorized ST-ConvNet [29]	71.3% (88.1%)
Two-stream + LSTM [37]	82.6% (88.6%)
Two-stream fusion [6]	82.6% (92.5%)
Long-term temporal ConvNet [33]	82.4% (91.7%)
Key-volume mining CNN [39]	84.5% (93.1%)
ST-ResNet [4]	82.2% (93.4%)
TSN [36]	85.7% (94.0%)
CNN-based representation extractor + linear SVM	
C3D [31]	82.3%
ResNet-152	83.5%
P3D ResNet	**88.6%**
Method fusion with IDT	
IDT [34]	85.9%
C3D + IDT [31]	90.4%
TDD + IDT [35]	91.5%
ResNet-152 + IDT	92.0%
P3D ResNet + IDT	**93.7%**

the accuracy of P3D ResNet can achieve 88.6%, making the absolute improvement over the best competitor TSN on the only frame input and ResNet-152 in the first and second category by 2.9% and 5.1%, respectively. Compared to [37] which operates LSTM over high-level representations of frames to explore temporal information, P3D ResNet is benefited from the temporal connections throughout the whole architecture and outperforms [37]. P3D ResNet with only frame input is still superior to [25, 29, 37] when fusing the results on the inputs of both frame and optical flow. The results also consistently indicate that fusing two kinds of inputs (performances in brackets) leads to apparent improvement compared to only using video frames. This motivates us to learn P3D ResNet architecture with other types of inputs in our future works. Furthermore, P3D ResNet utilizing 2D spatial convolutions plus 1D temporal convolutions exhibits significantly better performance than C3D which directly uses 3D spatio-temporal convolutions. By combining with IDT [34] which are hand-crafted features, the performance will boost up to 93.7%. In addition, by performing the recent state-of-the-art encoding method [22] on the activations of res5c layer in P3D ResNet, the accuracy can achieve 90.5%, making the improvement over the global representation from pool5 layer in P3D ResNet by 1.9%.

Table 4. Performance comparisons in terms of Top-1&Top-3 classification accuracy, and mean AP on ActivityNet validation set. A linear SVM classifier is learnt on each feature.

Method	**Top-1**	**Top-3**	**MAP**
IDT [34]	64.70%	77.98%	68.69%
C3D [31]	65.80%	81.16%	67.68%
VGG_19 [26]	66.59%	82.70%	70.22%
ResNet-152 [7]	71.43%	86.45%	76.56%
P3D ResNet	**75.12%**	**87.71%**	**78.86%**

Table 5. Action similarity labeling performances on ASLAN benchmark. STIP: Space-Time Interest Points; MIP: Motion Interchange Patterns; FV: Fisher Vector.

Method	**Model**	**Accuracy**	**AUC**
STIP [13]	linear	60.9%	65.3%
MIP [12]	metric	65.5%	71.9%
IDT+FV [19]	metric	68.7%	75.4%
C3D [31]	linear	78.3%	86.5%
ResNet-152 [7]	linear	70.4%	77.4%
P3D ResNet	linear	**80.8%**	**87.9%**

The results across different evaluation metrics constantly indicate that video representation produced by our P3D ResNet attains a performance boost against baselines on ActivityNet validation set, as shown in Table 4. Specifically, P3D ResNet outperforms IDT, C3D, VGG_19 and ResNet-152 by 10.4%, 9.3%, 8.5% and 3.7% in terms of Top-1 accuracy, respectively. There is also a large performance gap between C3D and ResNet-152. This is mainly due to data shift that the categories in ActivityNet are mostly human activities in daily life, which are quite different from those sport-related data in Sports-1M benchmark, resulting in not satisfying performance by C3D learnt purely on Sports-1M data. Instead, ResNet-152 trained on ImageNet image data is found to be more helpful in this case. P3D ResNet which pre-trains 2D spatial convolutions on image data and learns 1D temporal convolutions on video data fully leverages the knowledge from two domains, successfully boosting up the performance.

The second task is action similarity labeling challenge, which is to answer a binary question of “does a pair of videos present the same action?” Following the experimental settings in [13, 31], we extract the outputs of four layers in P3D ResNet, i.e., prob, pool5, res5c and res4b35 layer as four types of representation for each 16-frame video clip. The video-level representation is then obtained by averaging all clip-level representations. Given each video pair, we calculate 12 different similarities on each type of video representation and thus generate a 48-dimensional vector for each pair. An L2 normalization is implemented on the 48-d vector and a binary classifier is trained by using linear SVM. The performance comparisons on ASLAN are shown in Table 5. Overall, P3D ResNet performs consistently better than both hand-crafted features and CNN-based representations across the performance metric of accuracy and area under ROC curve (AUC). In general, CNN-based representations exhibits better accuracy than hand-crafted fea-

Table 6. The accuracy performance of scene recognition on Dynamic Scene and YUPENN sets.

Method	Dynamic Scene	YUPENN
[3]	43.1%	80.7%
[5]	77.7%	96.2%
C3D [31]	87.7%	98.1%
ResNet-152 [7]	93.6%	99.2%
P3D ResNet	**94.6%**	**99.5%**

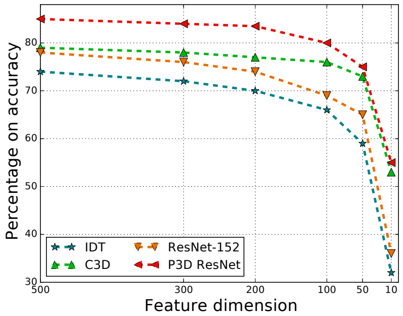
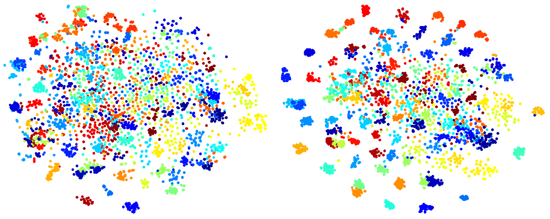


Figure 6. The accuracy of video representation learnt by different architectures with different dimensions. The performances reported in this figure are on UCF101 (3 splits).

tures. Unlike the observations on action recognition task, C3D significantly outperforms ResNet-152 on the scenario of action similarity labeling. We speculate that this may be the result of difficulty in interpreting the similarity between videos based on the ResNet-152 model learnt purely on image domain. In contrast, the video representation extracted by C3D which is trained on video data potentially has higher capability to distinguish between videos. At this point, improvements are also observed in P3D ResNet. This again indicates that P3D ResNet is endowed with the advantages of both C3D and ResNet-152 by pre-training 2D spatial convolutions on image data and learning 1D temporal connections on video data.

The third experiment was conducted on scene recognition. Table 6 shows the accuracy of different methods. P3D ResNet outperforms the state-of-the-art hand-crafted features [5] by 16.9% and 3.3% on Dynamic Scene and YUPENN benchmark, respectively. Compared to C3D and ResNet-152, P3D ResNet makes the absolute improvements by 1.4% and 0.3% on YUPENN, respectively.

The effect of representation dimension. Figure 6 compares the accuracy of video representation with different dimensions on UCF101 by performing Principal Components Analysis on the original features of IDT, ResNet-152, C3D and P3D ResNet. Overall, video representation learnt by P3D ResNet consistently outperforms others at each dimension from 500 to 10. In general, higher dimensional representation provide better accuracy. An interesting observation is that the performance of ResNet-152 decreases more sharply than that of C3D and P3D ResNet when reducing the representation dimension. This somewhat reveals the weakness of ResNet-152 in generating video representation, which is originated from domain gap that ResNet-152 is learnt purely on image data and may degrade the representational capability on videos especially when the feature dimension is very low. P3D ResNet, in comparison, is benefited from the exploration of knowledge from both image and video domain, making the learnt video representation more robust to the change of dimension.



(a) ResNet-152 (b) P3D ResNet

Figure 7. Video representation embedding visualizations of ResNet-152 and P3D ResNet on UCF101 using t-SNE [32]. Each video is visualized as one point and colors denote different actions.

Video representation embedding visualization. Figure 7 further shows the t-SNE [32] visualization of embedding of video representation learnt by ResNet-152 and P3D ResNet. Specifically, we randomly select 10K videos from UCF101 and the video-level representation is then projected into 2-dimensional space using t-SNE. It is clear that video representations by P3D ResNet are better semantically separated than those of ResNet-152.

6. Conclusion

We have presented Pseudo-3D Residual Net (P3D ResNet) architecture which aims to learn spatio-temporal video representation in deep networks. Particularly, we study the problem of simplifying 3D convolutions with 2D filters on spatial dimension plus 1D temporal connections. To verify our claim, we have devised variants of bottleneck building blocks for combining the 2D spatial and 1D temporal convolutions, and integrated them into a residual learning framework at different placements for structural diversity purpose. The P3D ResNet architecture learnt on Sports-1M dataset validate our proposal and analysis. Experiments conducted on five datasets in the context of video action recognition, action similarity labeling and scene recognition also demonstrate the effectiveness and generalization of the spatio-temporal video representation produced by our P3D ResNet. Performance improvements are clearly observed when comparing to other feature learning techniques.

Our future works are as follows. First, attention mechanism will be incorporated into our P3D ResNet for further enhancing representation learning. Second, an elaborated study will be conducted on how the performance of P3D ResNet is affected when increasing the frames in each video clip in the training. Third, we will extend P3D ResNet learning to other types of inputs, e.g., optical flow or audio.

References

[1] Deep draw. https://github.com/auduno/deepdraw. 6

[2] F. Caba Heilbron, V. Escorcia, B. Ghanem, and J. Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *CVPR*, 2015. 6

[3] K. Derpanis, M. Lecce, K. Daniildis, and R. Wildes. Dynamic scene understanding: The role of orientation features in space and time in scene classification. In *CVPR*, 2012. 6, 8

[4] C. Feichtenhofer, A. Pinz, and R. Wildes. Spatiotemporal residual networks for video action recognition. In *NIPS*, 2016. 7

[5] C. Feichtenhofer, A. Pinz, and R. P. Wildes. Bags of spacetime energies for dynamic scene recognition. In *CVPR*, 2014. 8

[6] C. Feichtenhofer, A. Pinz, and A. Zisserman. Convolutional two-stream network fusion for video action recognition. In *CVPR*, 2016. 2, 7

[7] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 1, 3, 4, 5, 7, 8

[8] S. Ji, W. Xu, M. Yang, and K. Yu. 3d convolutional neural networks for human action recognition. *IEEE Trans. on PAMI*, 35, 2013. 1, 2, 3

[9] Y.-G. Jiang, Z. Wu, J. Wang, X. Xue, and S.-F. Chang. Exploiting feature and class relationships in video categorization with regularized deep neural networks. *IEEE Trans. on PAMI*, 2017. 2

[10] 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, 5

[11] A. Klaser, M. Marszałek, and C. Schmid. A spatio-temporal descriptor based on 3d-gradients. In *BMVC*, 2008. 2

[12] O. Kliper-Gross, Y. Gurovich, T. Hassner, and L. Wolf. Motion interchange patterns for action recognition in unconstrained videos. In *ECCV*, 2012. 7

[13] O. Kliper-Gross, T. Hassner, and L. Wolf. The action similarity labeling challenge. *IEEE Trans. on PAMI*, 34(3), 2012. 6, 7

[14] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012. 5

[15] I. Laptev. On space-time interest points. *International journal of computer vision*, 64(2-3):107–123, 2005. 2

[16] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *CVPR*, 2008. 2

[17] Q. Li, Z. Qiu, T. Yao, T. Mei, Y. Rui, and J. Luo. Action recognition by learning deep multi-granular spatio-temporal video representation. In *ICMR*, 2016. 2

[18] Q. Li, Z. Qiu, T. Yao, T. Mei, Y. Rui, and J. Luo. Learning hierarchical video representation for action recognition. *International Journal of Multimedia Information Retrieval*, pages 1–14, 2017. 2

[19] X. Peng, Y. Qiao, Q. Peng, and Q. Wang. Large margin dimensionality reduction for action similarity labeling. *IEEE Signal Processing Letters*, 21(8):1022–1025, 2014. 7

[20] F. Perronnin, J. Sánchez, and T. Mensink. Improving the fisher kernel for large-scale image classification. In *ECCV*, 2010. 2

[21] Z. Qiu, Q. Li, T. Yao, T. Mei, and Y. Rui. Msr asia msm at thumos challenge 2015. In *CVPR workshop*, 2015. 2

[22] Z. Qiu, T. Yao, and T. Mei. Deep quantization: Encoding convolutional activations with deep generative model. In *CVPR*, 2017. 7

[23] P. Scovanner, S. Ali, and M. Shah. A 3-dimensional sift descriptor and its application to action recognition. In *ACM MM*, 2007. 2

[24] N. Shroff, P. Turaga, and R. Chellappa. Moving vistas: Exploiting motion for describing scenes. In *CVPR*, 2010. 6

[25] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In *NIPS*, 2014. 2, 7

[26] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. 7

[27] K. Soomro, A. R. Zamir, and M. Shah. UCF101: A dataset of 101 human action classes from videos in the wild. *CRCV-TR-12-01*, 2012. 4, 6

[28] N. Srivastava, E. Mansimov, and R. Salakhutdinov. Unsupervised learning of video representations using lstms. In *ICML*, 2015. 2

[29] L. Sun, K. Jia, D.-Y. Yeung, and B. E. Shi. Human action recognition using factorized spatio-temporal convolutional networks. In *ICCV*, 2015. 7

[30] 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. 5

[31] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3d convolutional networks. In *ICCV*, 2015. 1, 2, 3, 5, 7, 8

[32] L. van der Maaten and G. Hinton. Visualizing data using t-sne. *JMLR*, 2008. 8

[33] G. Varol, I. Laptev, and C. Schmid. Long-term temporal convolutions for action recognition. In *arXiv preprint arXiv:1604.04494*, 2016. 1, 2, 7

[34] H. Wang and C. Schmid. Action recognition with improved trajectories. In *ICCV*, 2013. 2, 7

[35] L. Wang, Y. Qiao, and X. Tang. Action recognition with trajectory-pooled deep-convolutional descriptors. In *CVPR*, 2015. 2, 7

[36] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. Van Gool. Temporal segment networks: towards good practices for deep action recognition. In *ECCV*, 2016. 2, 4, 6, 7

[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 *CVPR*, 2015. 2, 5, 7

[38] X. Zhang, Z. Li, C. C. Loy, and D. Lin. Polynet: A pursuit of structural diversity in very deep networks. *arXiv preprint arXiv:1611.05725*, 2016. 5

[39] W. Zhu, J. Hu, G. Sun, X. Cao, and Y. Qiao. A key volume mining deep framework for action recognition. In *CVPR*, 2016. 2, 7