Deep Dive into Convolutional 3D features for action and activity recognition (C3D)

Binu M. Nair Follow
Jul 23, 2018 · 9 min read

In this blog, I would provide a brief overview of the Convolutonal 3D or C3D model widely used in video recognition applications and recent challenges in Kaggle (ECCV YouTube 8M challenge). My motivation for this blog is that recently at CVPR (2018), I came across a number of models especially in action and activity recognition problems where C3D have been used as a de-facto feature extraction technique. So for those who are new or currently exploring computer vision especially in the video domain, this blog can be beneficial where I will explore its architecture and give some technical insights into this model. For readers interested in knowing concrete details about the experimentation, here is the reference given below.

> *Learning spatiotemporal features with 3d convolutional networks ; D Tran, L Bourdev, R Fergus, L Torresani, M Paluri , 2015 IEEE International Conference on Computer Vision (ICCV), 4489–4497*

What are C3D features?

C3D are deep 3-dimensional convolutional neural networks with a homogenous architecture containing 3 x 3 x 3 convolutional kernels followed by 2 x 2 x 2 pooling at each layer. They are trained on a large scale supervised video dataset such as UCF-101 and Sports 1M. The key properties that make it stand against traditional pre-trained models such as ResNets, AlexNet are the following:

1. **Generic** feature extraction: The 3D convolutions extracts both spatial and temporal components relating to motion of objects, human actions, human-scene or human-object interaction and appearance of those objects, humans and scenes. Thus it is not limited to appearance representation as in ResNets or AlexNet. This makes it a very

generic video feature representation for various video related tasks such as action localization, and event detection without the need for fine-tuning for each task.

2. **Compact** representation: The C3D model is given an input video segment of 16 frames (after downsampling to a fixed size which depends on dataset used) and the outputs a 4096-element vector. It gives you a compact representation of a video segment which can be further fed to either a temporal network for action localization task [3] , action recognition [1], and action-word mining [4]. Thus, it makes storage and retrieval of videos highly scalable.

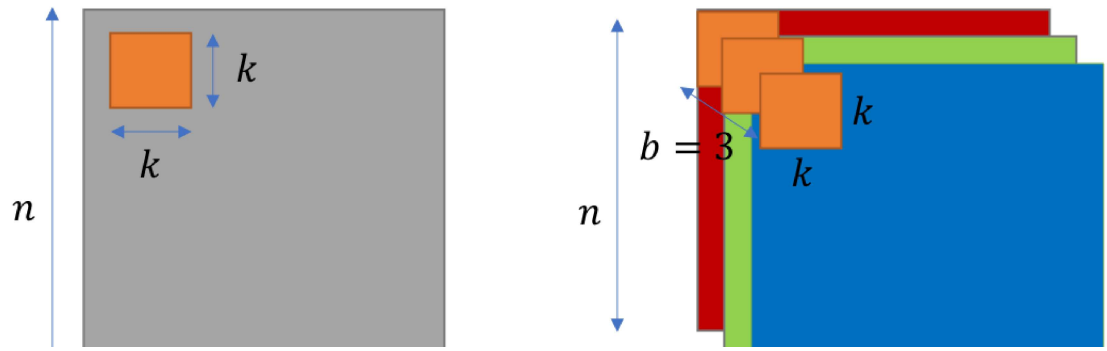
3. **Fast, efficient inference:** Due to its homogenous architecture with small kernel sizes of 3 x 3 x 3, it can facilitate for optimized implementation of these architectures for embedded platforms. This can have huge implications for smart camera environments where processing of real-time feed and extraction of meta data is critical for surveillance and smart city applications.

Before we dwell into the network architecture, I will focus first on the concept of 3D convolution on a spatial temporal cube of size b x L x n x n where c is the number of channels, L is the number of frames.

3D Convolution Operator — Basics

There are typically four types of convolution that we commonly see in computer vision:

1. 2D convolution on gray scale image
2. 2D convolution on RGB image
3. 2D convolution on L-frame RGB video segment
4. 3D convolution on L-frame RGB segment.



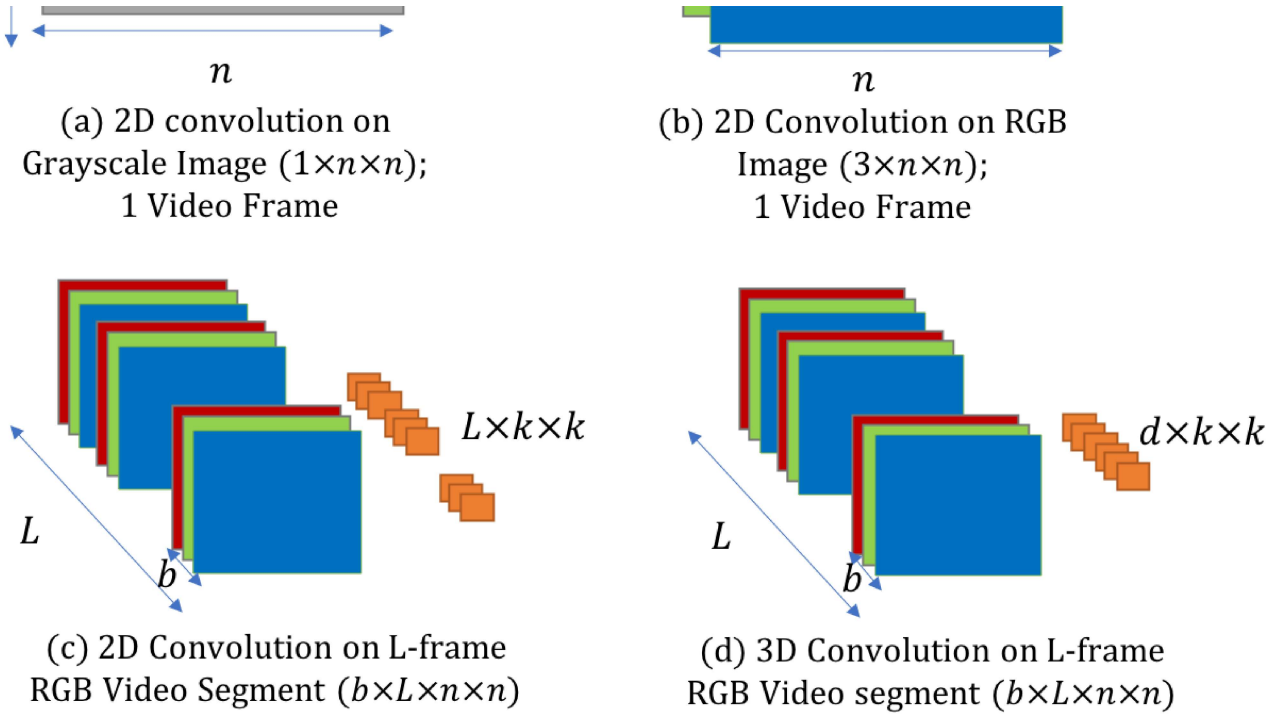
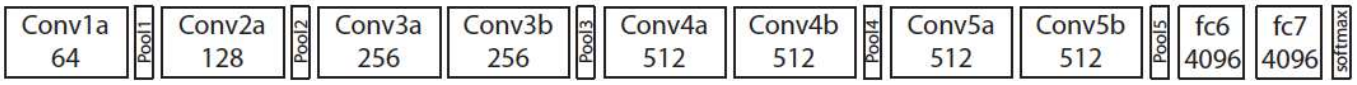


Illustration of 3D convolution on L-frame RGB video segment

A 2D convolution of an *n* x *n* image with a kernel of size *k* x *k* results in another 2D image. Similar is the case with the RGB image (*3* x *n* x *n*) where the kernel of size *3* x *k* x *k* is convolved with the image resulting in another 2D image (Figures (a),(b)). Now, when to comes to an *L*-frame RGB video segment, the effective input channel length will be *3* x *L*. Therefore, a 2D convolution operation will result in convolving with a kernel of size *3* x L x *k* x *k* resulting in again a 2D image (as shown sub figure c). The issue in a 2D convolution on a video segment is that the temporal features at this point gets squashed by this operation resulting in a temporally averaged appearance feature map without any motion representation. This can be circumvented by using a 3D convolution operation where we explicity define a kernel of size *d* x *k* x *k* where now, the convolution operation is between a RGB video segment of size *3* x *L* x *n* x *n* with an effective kernel of size *3* x *d* x *k* x *k* resulting in an output cuboid which preserves the temporal information.

C3D Architecture

The network architecture contains 8 convolutional, 5 pooling layers and 2 fully connected layers as shown in Figure below.



C3D Architecture [1]

The first convolution layer of size *1*x*3*x*3* followed by a pooling layer of size *1*x*2*x*2*. This is to preserve the temporal information in the first layer and build higher level representation of the temporal information at subsequent layers of the network. Every other convolution layer and pooling layer would have a size of *3*x*3*x*3* and *2*x*2*x*2* where the strides are 1 and 2 respectively. The fully connected layers have a size of *4096* dimensions with softmax outputs reflecting either 101 classes of UCF-101 dataset or 487 classes of the Sports 1M dataset. For those who wish to play with the C3D model, apre-trained model trained on Sports 1M dataset is available as open-source (link). An implementation in Keras which ports a trained model from Caffe is also available (link).

Key Takeaways

Temporal information representation

One of the key takeaways in this architecture design is the use of 3D convolutions which retains the temporal information across the feature maps for a video sequence. The figure below shows the deconvolution of the Conv5b feature maps with the highest activation. We see that in the first few frames, the features focus more on the appearance of the actor. As the action proceeds, the feature focus more on the salient motion near the pole.

Pole Vault Action : Emphasis on human appearance (Feature map from Conv5b for first few frames)[1]

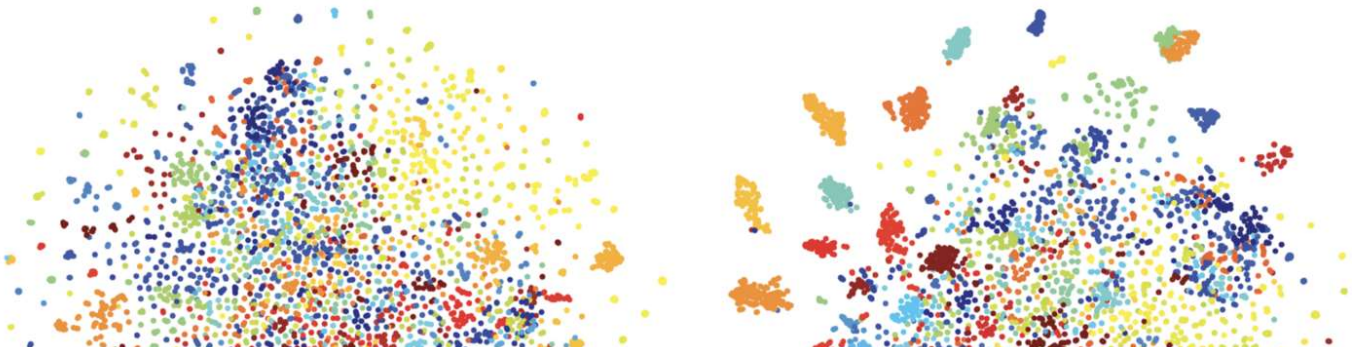
Pole Vault Action : Emphasis is on salient motion on pole (Feature map from Conv5b for last few frames)[1]

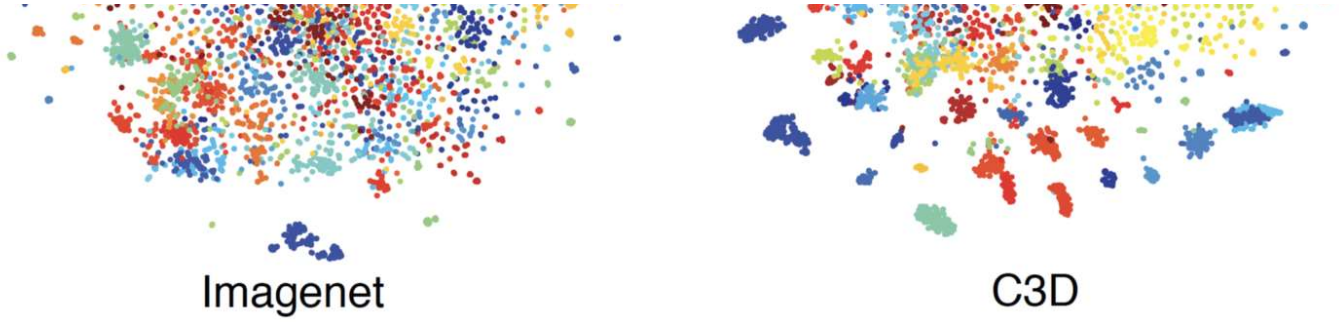
Similarly, for the action sequence “apply makeup” shown below, the first few frames focus on the facial appearance and the last few frames on the salient motion occuring near the eye brows and eyes. Thus, C3D filters selectively focus on appearance and motion at different instants of a video segment. This is the core differentiator with standard 2D ConvNets such as AlexNet or ResNet where the filters focus only on appearance and the temporal information averaged out across the frames.

Apply Makeup : Emphasis is on facial appearance (Feature map from Conv5b for first few frames)[1]

Apply Makeup : Emphasis is on salient motion near eyebrows (Feature map from Conv5b for last few frames)[1]

Better separability across action classes



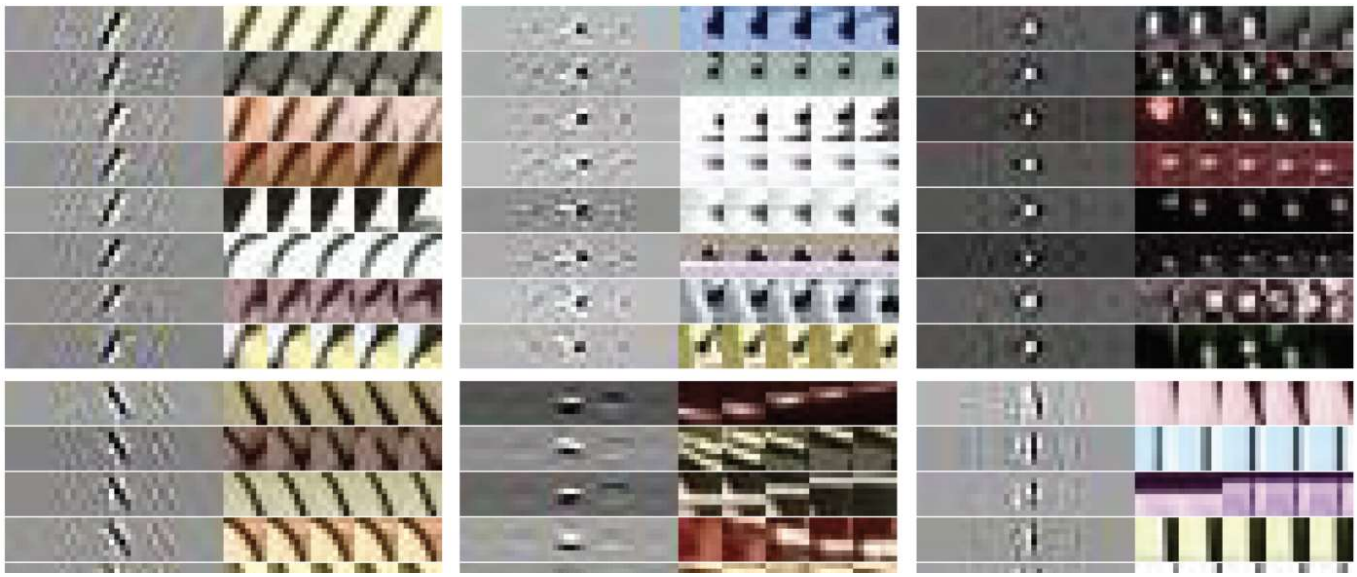


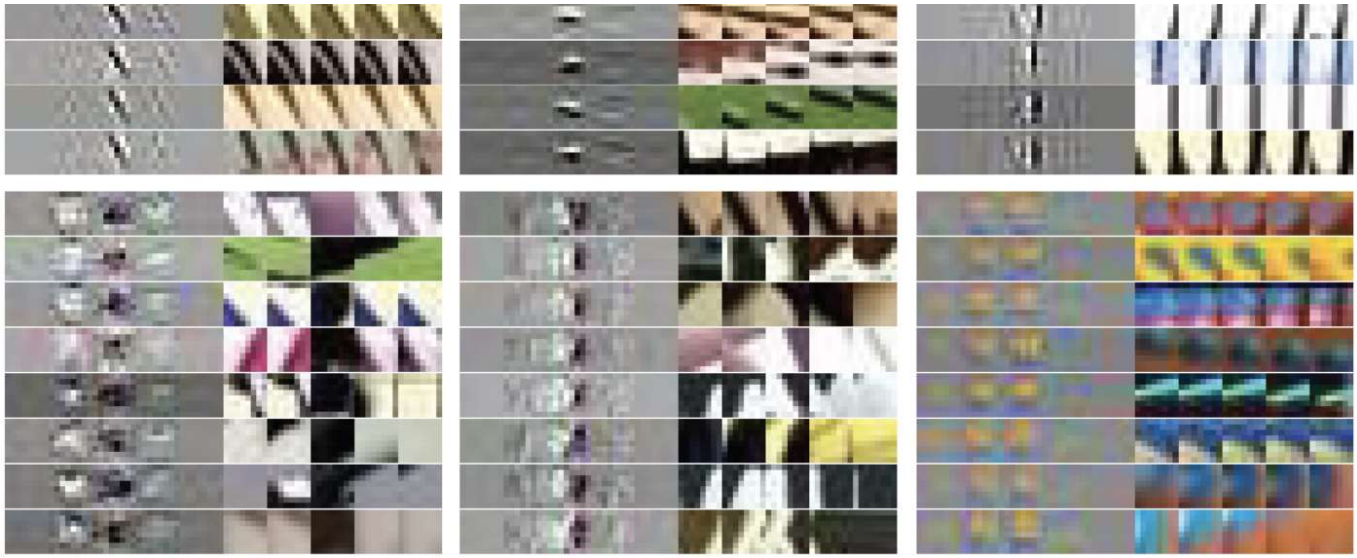
Video Feature representation in low dimensional space[1]

Another key takeway from this model is the genetic nature of the features for video representation. The figure above shows the distribution of the video features extracted from ImageNET-based network and the C3D network in low dimensional space. The video features represent the fully connected layer ouputs averaged across the 16-frame segments, and the 2-dimensional embedded space is obtained by applying t-SNE dimensionality reduction technique [2]. Note that these features are not assigned class labels and are projected in an unsupervised manner. We see that the features from C3D have much better seperability than those extracted from networks trained on ImageNet. This shows that for videos, C3D provides both a generic and separable representation for video sequences.

Spatio-temporal feature abstraction

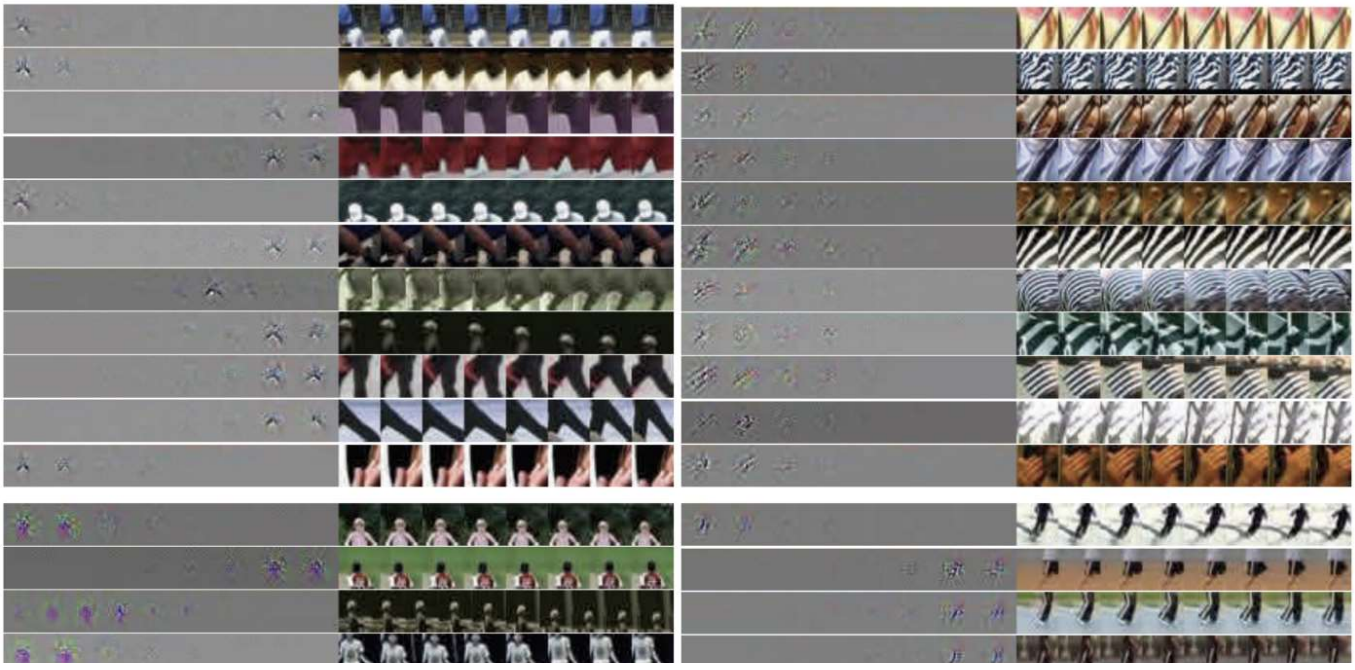
The C3D network also represents spatio-temporal information at different levels of abstraction ranging from detection of moving edges, tracking corner movements, and emphasis on people and objects relevant to the action. An illustration of the information learned by the first layer (Conv1b) is shown below.

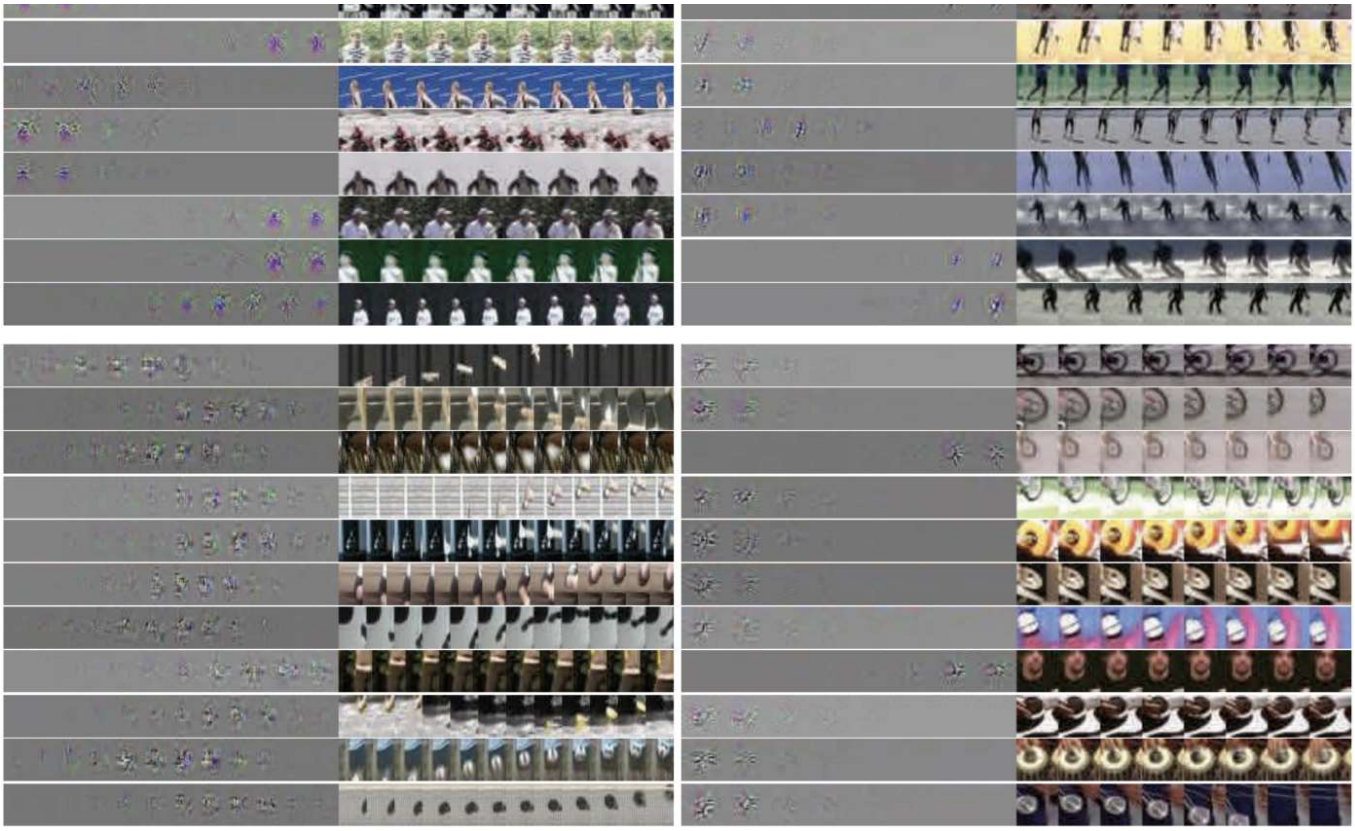




Filter visualization using DeConvolution method from Conv1b[1]

The set of filters in first two rows corresponds to moving edges and blob detectors of varying sizes, thickness and flexibility. We see that the deconvolution of such filters provides the kind of patches that gives the highest response for the filters in an image. Similarly, in the third row, the filters are more inclined to give highest responses to different color and texture variations. For example, in the last row and column of the figure above, the filters provide highest responses to image patches that are reddish or yellowish in color with some sutle responses to the bluish regions. We can also see the filters can detect specific combination of colors or a combination of a color and an edge (eg: blue planar region imposed on a black region). This particular type of filter representation provides only a single level of spatio-temporal feature abstraction.





Filter visualization using DeConvolution method from Conv3b[1]

In the figure above, a different level of spatio-temporal feature abstraction is provided and these are from the deconvolution outputs of Conv3b filters. In the first row, the filters detect moving smaller body parts such as arms and legs associated with the motion and texture like features corresponding to the context in the action. The second row represents filters which detects much larger body regions of the human performing various types of actions. In the third row and first column, the filters capture various movements associated with object trajectories. However in the second column, the filters characterize objects which are circular in nature and captures the context of an action. Thus, through these visualizations, we see that C3D network captures spatio-temporal features at various abstract levels which represents both motion and appearance (context) necessary to represent a video segment.

How and why is C3D important?

C3D is a very common video segment descriptor which embeds both motion and temporal characteristics. Thus, it is very intuitive to use C3D features as a first step before further processing for various applications.

1. Video representation [1]: For representing a long video of 2 sec, the video is broken down in 16-frame temporal chunks with 50% overlap between consecutive chunks. The C3D features are then averaged across the temporal segments/chunks using L-2 norm to give a robust compact description of the long video.

2. Action-Word Embedding [3] : Given a video segment, a network can be trained to model semantic relationship between a video feature and a corresponding word vector representation of the action class. This helps in building action word vectors. Using these action-word vectors, videos can be queried in an unsupervised manner based on this embedding and helps in zero-shot action recognition applications.

3. Temporal Localization [4] : A Faster R-CNN type architecture has been used to segment and localize video sequences into specific human actions in long continous videos. For this approach, C3D features are used along with multi temporal scale R-CNN model.

4. Video Generation [5] : For generating videos from a set of frames, C3D features are used during the comparison stage between the generated frames and true frames during training.

To conclude

In this article, I provide a quick overview of the C3D model and describe some of its applications in current research. In my next blog, I will provide details on the current state of art video descriptor known as I3D [6] where this model has been trained on a very large scale dataset known as Kinetics 600.

References

[1] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, ***Learning Spatiotemporal Features with 3D Convolutional Networks***, ICCV 2015

[2] Maaten, Laurens van der and Geoffrey E. Hinton. "***Visualizing Data using t-SNE.***" (2008).

[3] Meera Hahn, Andrew Silva, James M. Reh, "***Action2Vec: A Crossmodal Embedding Approach to Zero Shot Action Learning***", Computer Vision and Pattern Recognition (CVPR) 2018 Deep Vision Workshop

[4] Y. W. Chao, S. Vijayanarasimhan, B. Seybold, D. Ross, J. Deng, R. Sukthankar. "***Rethinking the Faster R-CNN Architecture for Temporal Action Recognition***". Computer Vision and Pattern Recognition (CVPR), 2018

[5] De-An Huang, Vignesh Ramanathan, Dhruv Mahajan, Lorenzo Torresani, Manohar Paluri, Li Fei-Fei, Juan Carlos Niebles, "***What Makes a Video a Video: Analyzing Temporal Information in Video Understanding Models and Dataset***", Computer Vision and Pattern Recognition (CVPR) 2018

[6] J Carreira, A Zisserman, "***Quo vadis, action recognition? a new model and the kinetics dataset"*** Computer Vision and Pattern Recognition (CVPR), 2017

Computer Vision Deep Learning Video Recognition Activity Recognition