一、Two-stream

1 、Two-Stream Convolutional Networks for Action Recognition in Videos

http://www.robots.ox.ac.uk/~vgg/software/two_stream_action/

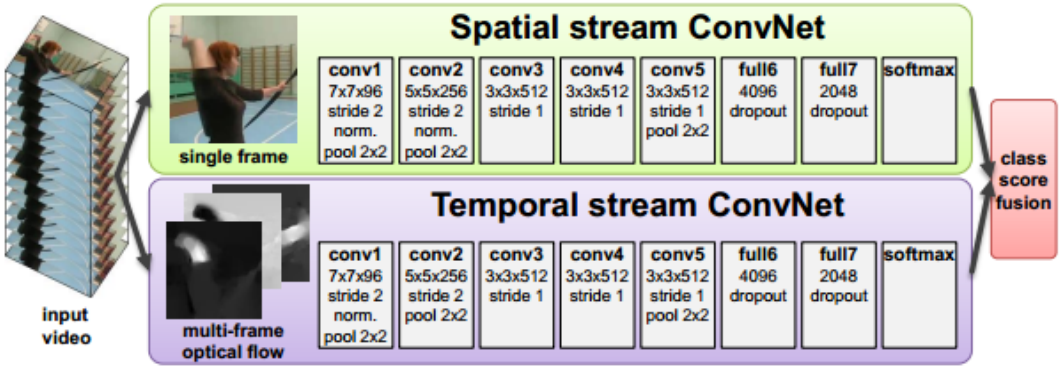
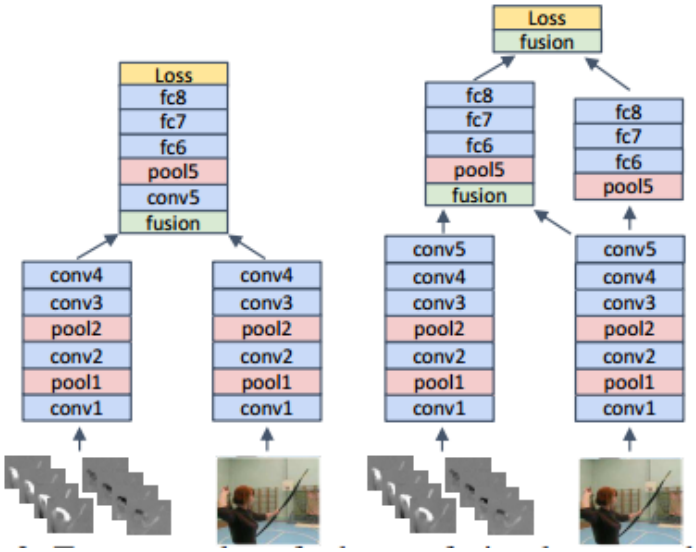


Figure 1: **Two-stream architecture for video classification.**

14 年提出双流，利用帧图像和光流图像作为 CNN 的输入得到很好的效果。光流能够描述出视频帧的运动信息，一路是连续几帧的光流叠起来作为 CNN 的输入；另一路就是普通的单帧的 CNN。其实就是两个独立的神经网络了，最后再把两个模型的结果平均一下。另外，它利用 multi-task learning 来克服数据量不足的问题。其实就是 CNN 的最后一层连到多个 softmax 的层上，对应不同的数据集，这样就可以在多个数据集上进行 multi-task learning。

2、 Convolutional Two-Stream Network Fusion for Video Action Recognition

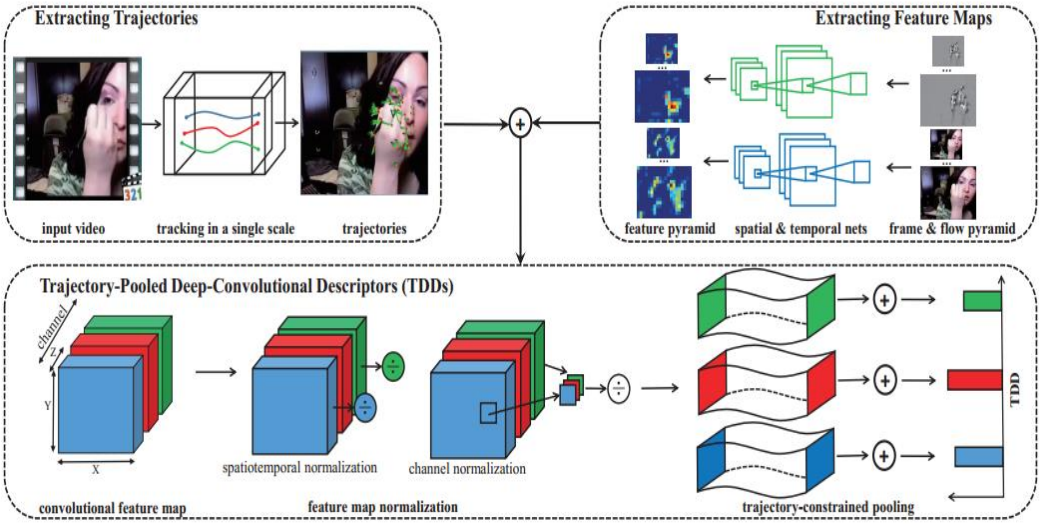
http://www.robots.ox.ac.uk/~vgg/software/two_stream_action/
https://github.com/feichtenhofer/twostreamfusion



16 年针对双流融合问题进行研究，得到卷积结束之后在全链接之前融合效果比较好，左边是单纯在某一层融合，右边是融合之后还保留一路网络，在最后再把结果融合一次。论文的实验表明，后者的准确率要稍高。

3、 Action Recognition with Trajectory-Pooled Deep-Convolutional Descriptors

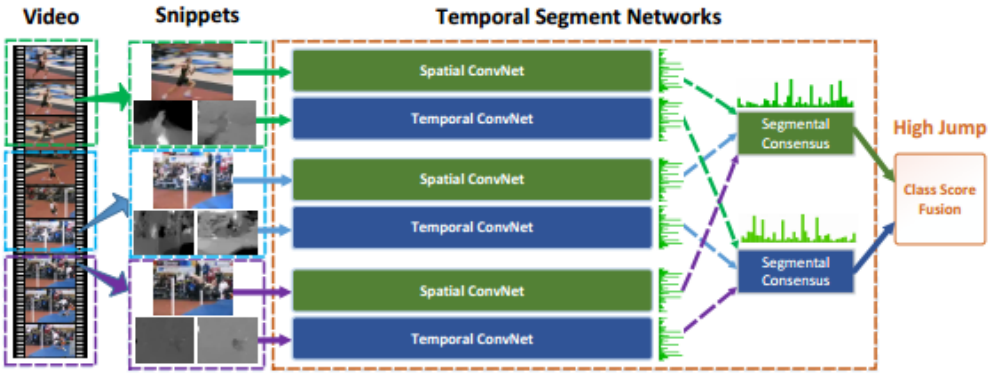
https://wanglimin.github.io/tdd/index.html



论文考虑了时间维的特性，引进了轨迹控制策略来采样，将手工设计的特征和深度学习结合。首先多个空间尺度上密集采样特征点，然后特征点跟踪得到轨迹形状特征，同时需要更有力的特征来描述光流，Fisher Vector 方法进行特征的编码，最后 svm 采用 one-against-rest 策略训练多类分类器。

4、 Temporal Segment Networks: Towards Good Practices for Deep Action Recognition

https://github.com/yjxiong/temporal-segment-networks

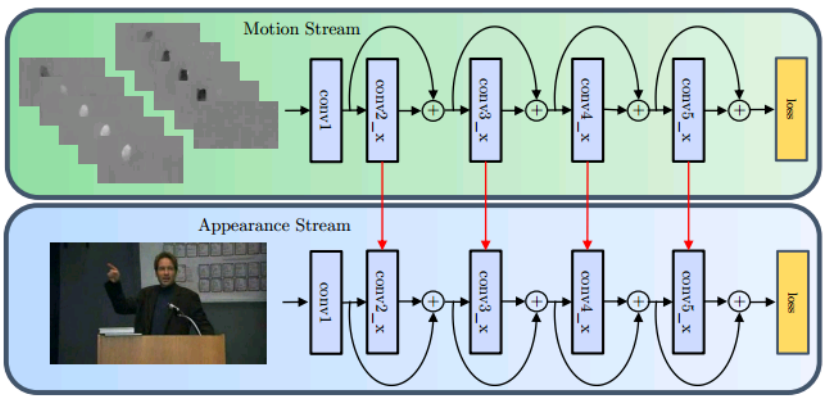


16 年香港中文大学针对双流不能很好利用长时间信息，提出 segment 思路，将视屏分为前中后三段，每段经过双流然后融合结果。其中港中文还做了很多其他工作，https://arxiv.org/abs/1507.02159 比较了各种网络在 action recognition 中的效果，https://wanglimin.github.io/ http://yjxiong.me/ 膜拜大神们

HMDB51		UCF101	
DT+MVSV [37]	55.9%	DT+MVSV [37]	83.5%
iDT+FV [2]	57.2%	iDT+FV [38]	85.9%
iDT+HSV [25]	61.1%	iDT+HSV [25]	87.9%
MoFAP [39]	61.7%	MoFAP [39]	88.3%
Two Stream [1]	59.4%	Two Stream [1]	88.0%
VideoDarwin [18]	63.7%	C3D (3 nets) [13]	85.2%
MPR [40]	65.5%	Two stream +LSTM [4]	88.6%
$F_{ST}CN$ (SCI fusion) [28]	59.1%	$F_{ST}CN$ (SCI fusion) [28]	88.1%
TDD+FV [5]	63.2%	TDD+FV [5]	90.3%
LTC [19]	64.8%	LTC [19]	91.7%
KVMF [41]	63.3%	KVMF [41]	93.1%
TSN (2 modalities)	68.5%	TSN (2 modalities)	94.0%
TSN (3 modalities)	**69.4%**	TSN (3 modalities)	**94.2%**

5、 Spatiotemporal Residual Networks for Video Action Recognition

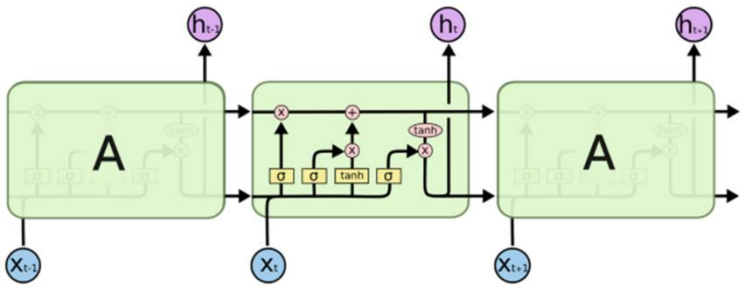
https://feichtenhofer.github.io/



使用了两个流，但是名字不是取为空间流和时间流，而是 motion stream 和 appearance stream，但是本质不变，运动流接收的输入依然是堆叠的多帧光流灰度图片，而 appearance stream 和原来的空间流一致，接收的输入都是 RGB 图片，但是这里使用的 双流的两个流之间是有数据交换的，而不是像 TSN 网络一样在最后的得分进行融。单帧的潜力挖尽之后自然就会有人上 3D Conv，Recurrent CNN，Grid RNN 之类的东西。虽然深度学习大法好，不过也得按基本法来，直接上 fancy 的模型有较大概率吃力不讨好。

Method	UCF101	HMDB51	Method	UCF101	HMDB51
Two-Stream ConvNet [20]	88.0%	59.4%	IDT [29]	86.4%	61.7%
Two-Stream+LSTM[18]	88.6%	-	C3D + IDT [26]	90.4%	-
Two-Stream (VGG16) [1, 31]	91.4%	58.5%	TDD + IDT [30]	91.5%	65.9%
Transformations[31]	92.4%	62.0%	Dynamic Image Networks + IDT [2]	89.1%	65.2%
Two-Stream Fusion[5]	92.5%	65.4%	Two-Stream Fusion[5]	93.5%	69.2%
ST-ResNet*	**93.4%**	**66.4%**	ST-ResNet* + IDT	**94.6%**	**70.3%**

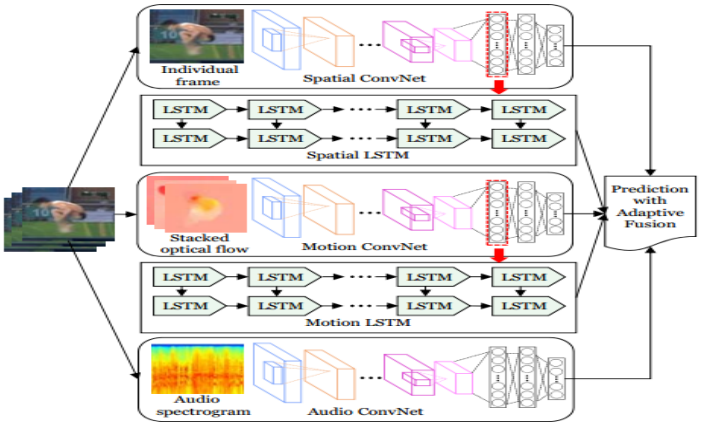
二 、LSTM 结构



Long Short Term 网络一般就叫 LSTM，它是一种 RNN 特殊的类型。LSTM 通过刻意的设计来避免长期依赖问题。记住长期的信息在实践中是 LSTM 的默认行为，而非付出很大代价才能获得的能力！

1、 Fusing Multi-Stream Deep Networks for Video Classification

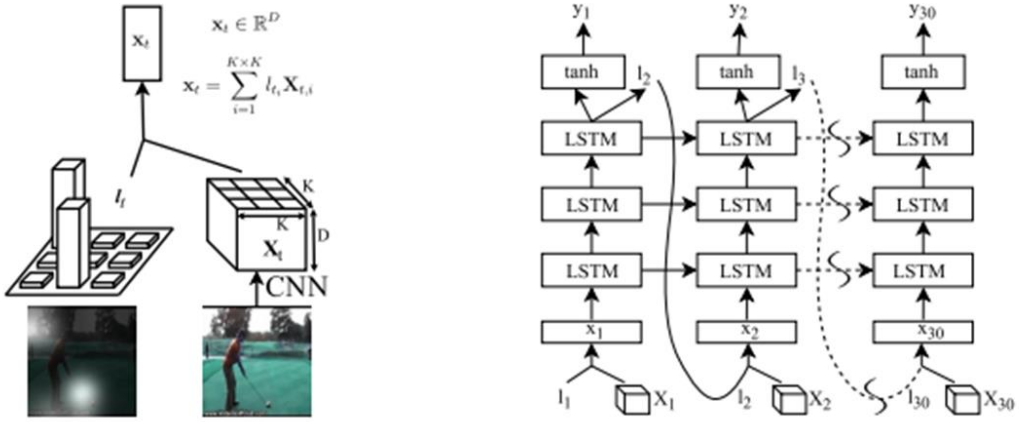
https://arxiv.org/abs/1509.06086



文章先 CNN 提取特征，包括 rgb 图光流图和语音频谱图，然后经过 lstm 最后融合。

2、 Action Recognition using Visual Attention

http://shikharsharma.com/projects/action-recognition-attention/
http://www.cs.cmu.edu/~rsalakhu/

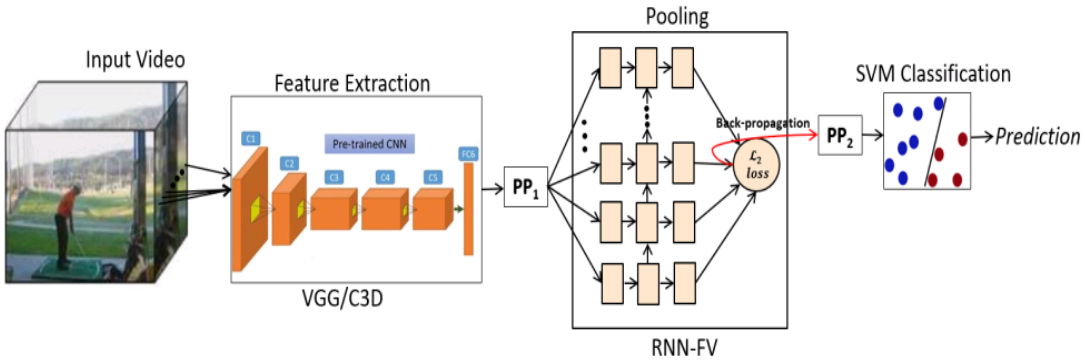


(a) The soft attention mechanism　　(b) Our recurrent model

注意力模型终于来啦，人在看东西的时候，目光沿感兴趣的地方移动，甚至仔细盯着部分细节看，然后再得到结论。Attention 就是在网络中加入关注区域的移动、缩放机制，连续部分信息的序列化输入。采用 attention 使用时间很深的 lstm 模型，学习视屏的关键运动部位。Attention 相关：
http://www.cosmosshadow.com/ml/%E7%A5%9E%E7%BB%8F%E7%BD%91%E7%BB%9C/2016/03/08/Attention.html

3、 RNN Fisher Vectors for Action Recognition and Image Annotation

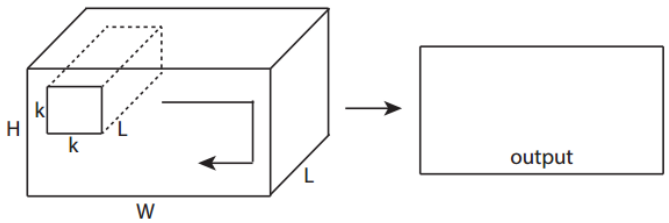
http://www.eccv2016.org/files/posters/P-4A-30.pdf



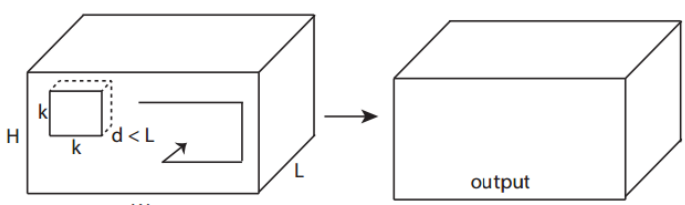
文章典型的特征提取，分类思路文章采用卷积网络提取特征之后进过 pca 降维，然后 Fisher Vector 编码扔给 RNN 再 pca 降维，最后 svm 分类。Ucf101 上实验结果到了 94%

Method	HMDB51	UCF101
idt [49]	57.2	85.9
idt + high-D encoding [53]	61.1	87.9
Two-Stream CNN (2 nets) [28]	59.4	88
Multi-Skip feature stacking [54]	65.4	89.1
C3D (1 net) [14] C3D (3 nets) [14] C3D (3 nets) + idt [14]	- - -	82.3 85.2 90.4
TDD (2 nets) [5] TDD (2 nets) + idt [5]	63.2 65.9	90.3 91.5
Stacked FV [2] Stacked FV + idt [2]	56.21 66.78	- -
RNN-FV (C3D + VGG-CCA) RNN-FV (C3D + VGG-CCA) + idt	54.33 **67.71**	88.01 **94.08**

三、C3D



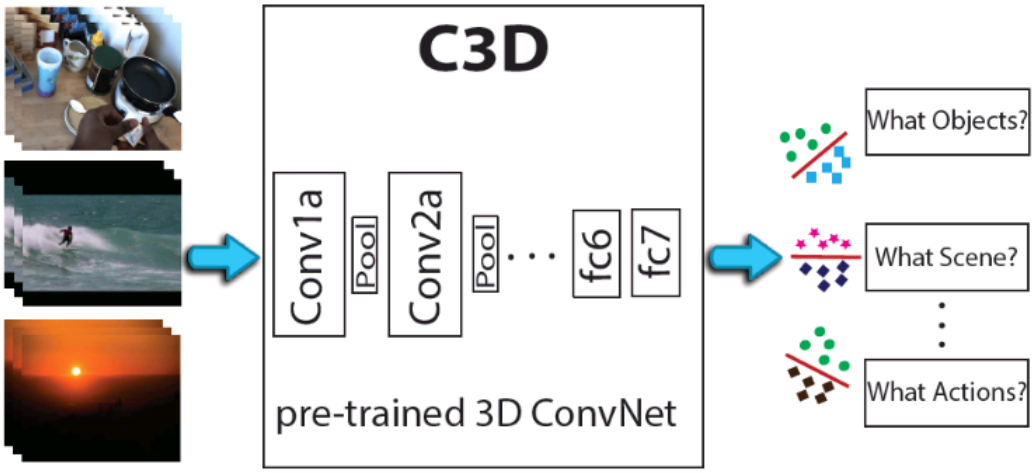
2D convolution on multiple frames



3D CNN 应用于一个视频帧序列图像集合，并不是简单地把图像集合作为多通道来看待输出多个图像（这种方式在卷积和池化后就丢失了时间域的信息，如图 6 上），而是让卷积核扩展到时域，卷积在空域和时域同时进行，输出仍然是有机的图像集合

1、Learning Spatiotemporal Features with 3D Convolutional Networks

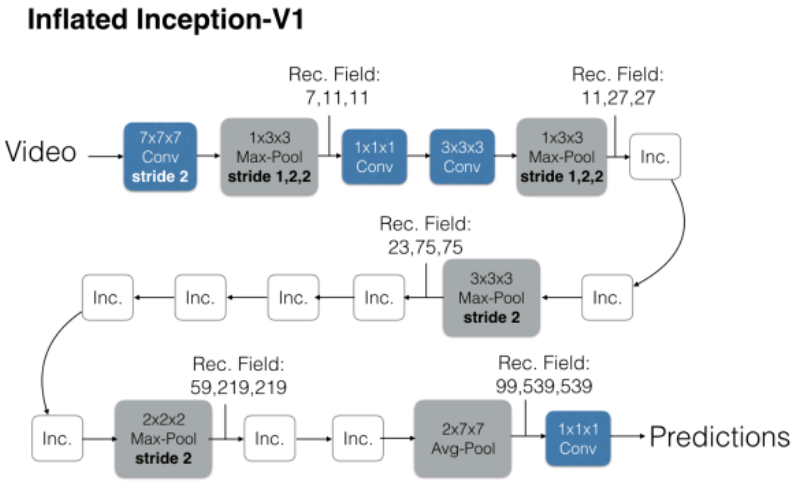
https://github.com/facebook/C3D
https://gist.github.com/albertomontesg/d8b21a179c1e6cca0480ebdf292c34d2



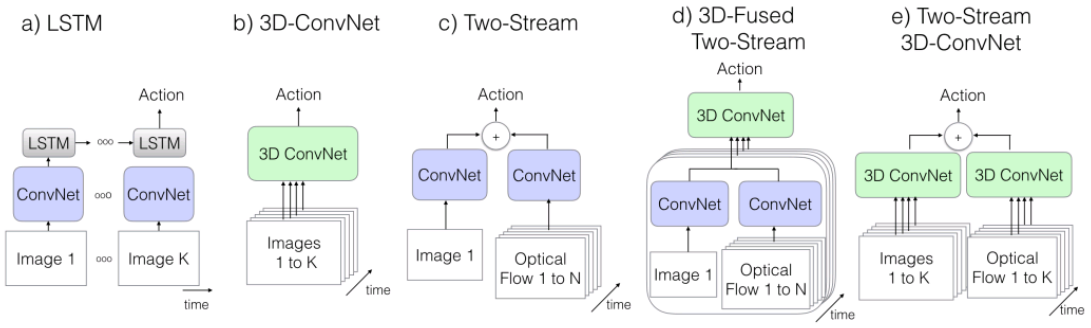
将视频分成多个包含 16 帧的片段作为网络的输入。第一个池化层 d=1，是为了保证时间域的信息不要过早地被融合，接下来的池化层的 d=2。有所卷积层的卷积核大小为 3x3x3，相对其他尺寸的卷积核，达到了精度最优，计算性能最佳。

2、 Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset

https://deepmind.com/research/publications/quo-vadis-action-recognition-new-model-and-kinetics-dataset/



以往的 Conv3D 效果很差的原因之一就是数据集太小，喂不饱网络。文章中的 3D 网络并不是随机初始化的，而是将在 ImageNet 训好的 2D 模型参数展开成 3D，之后再训练。因此叫 Inflating 3D ConvNets. 本文选用的网络结构为 BN-Inception(TSN 也是)，但做了一些改动。如果 2D 的滤波器为 N*N 的，那么 3D 的则为 N*N*N 的。具体做法是沿着时间维度重复 2D 滤波器权重 N 次，并且通过除以 N 将它们重新缩放. 在前两个池化层上将时间维度的步长设为了 1，空间还是 2*2。最后的池化层是 2*7*7。训练的时候将每一条视频采样 64 帧作为一个样本，测试时将全部的视频帧放进去最后 average_score。除最后一个卷积层之外，在每一个都加上 BN 层和 Relu。对于 I3D 的效果为什么好，作者解释说 I3D 有 64 帧的感受野。可以更好地学习时序信息。再就是先用 ImageNet 的模型做了预训练



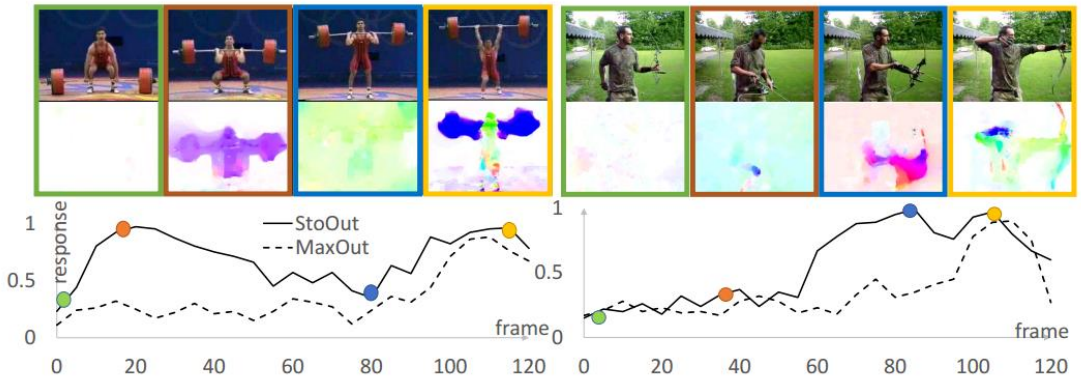
Architecture	UCF-101			HMDB-51			miniKinetics		
	RGB	Flow	RGB + Flow	RGB	Flow	RGB + Flow	RGB	Flow	RGB + Flow
(a) LSTM	81.0	–	–	36.0	–	–	69.9	–	–
(b) 3D-ConvNet	51.6	–	–	24.3	–	–	60.0	–	–
(c) Two-Stream	83.6	85.6	91.2	43.2	56.3	58.3	70.1	58.4	72.9
(d) 3D-Fused	83.2	85.8	89.3	49.2	55.5	56.8	71.4	61.0	74.0
(e) Two-Stream I3D	**84.5**	**90.6**	**93.4**	**49.8**	**61.9**	**66.4**	**74.1**	**69.6**	**78.7**

Model	UCF-101	HMDB-51
Two-Stream [25]	88.0	59.4
IDT [30]	86.4	61.7
Dynamic Image Networks + IDT [2]	89.1	65.2
TDD + IDT [31]	91.5	65.9
Two-Stream Fusion + IDT [8]	93.5	69.2
Temporal Segment Networks [32]	94.2	69.4
ST-ResNet + IDT [7]	94.6	70.3
Deep Networks [15], Sports 1M pre-training	65.2	-
C3D one network [29], Sports 1M pre-training	82.3	-
C3D ensemble [29], Sports 1M pre-training	85.2	-
C3D ensemble + IDT [29], Sports 1M pre-training	90.1	-
RGB-I3D, miniKinetics pre-training	91.8	66.4
RGB-I3D, Kinetics pre-training	95.6	74.8
Flow-I3D, miniKinetics pre-training	94.7	72.4
Flow-I3D, Kinetics pre-training	96.7	77.1
Two-Stream I3D, miniKinetics pre-training	96.9	76.3
Two-Stream I3D, Kinetics pre-training	**98.0**	**80.7**

I3D 这个网络结构的提出是很显然，但用 2D 的 ImageNet 模型做预训练以及参数展开分配还是挺具有创新性的，虽然在 TSN 中处理光流的第一个卷积层时就有使用过类似的方法。这个实验室真有能力，以往的数据集上效果很难提升，自己就搞了个大数据集。那个 Kinetics 的 I3D 模型是在 **64 块 GPU** 上跑出来的。

四 、其他

1、 A Key Volume Mining Deep Framework for Action Recognition



现即便是 trimmed video （例如 UCF101 数据集），实际的动作发生的时空位置也是非常不确定的：我们既不知道做动作的人在什么空间位置，也不知道真正的动作发生的精确时间位置。更糟糕的是，和动作类别直接相关的，具有区分性的 （discriminative） key volume 往往占比非常小，这在 flow stream 上表现得尤为突出。

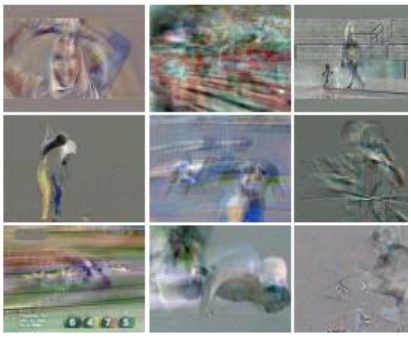
于是我们就想能否先把这些 key volume 找出来，直接用以训练分类器，这样可以免受噪声数据的干扰，更加聚焦在动作本质上。但实际上，在得到一个好的分类器之前我们是很难自动地将 key volume 挑出来的。于是我们陷入了一个鸡生蛋，蛋生鸡的困境。

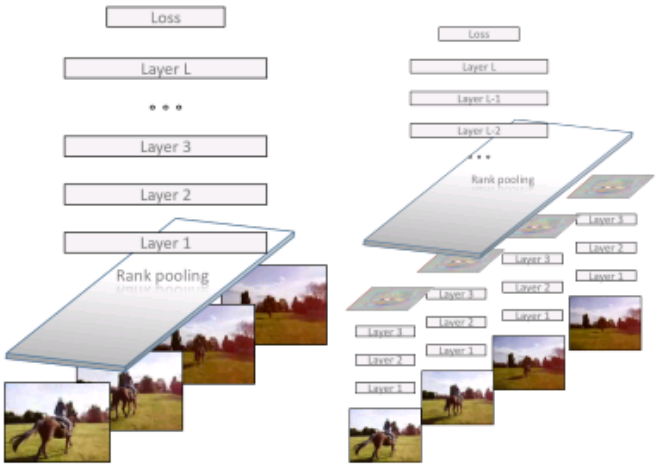
借鉴 Multiple Instance Learning 的思想，我们把鸡和蛋的问题放在一起来优化解决：在训练分类器的同时，挑选 key volume；并用挑出来的 key volume 更新分类器的参数。这两个过程无缝地融合到了 CNN （卷积神经网络）的网络训练的 forward 和 backward

过程中，使得整个训练过程非常优雅、高效。

2 、 Dynamic Image Networks for Action Recognition

https://github.com/hbilen/dynamic-image-nets





CNN 的输入一般是图片，针对输入

表征视屏的信息？答案是可疑的，针对对视频中的 RGB 图像进行 rank pooling 处理，以此作为 cnn 的输入。虽然最终的效果不是特别好，但是想法很 nice。