

行为识别

Deep Learning

逐帧处理融合

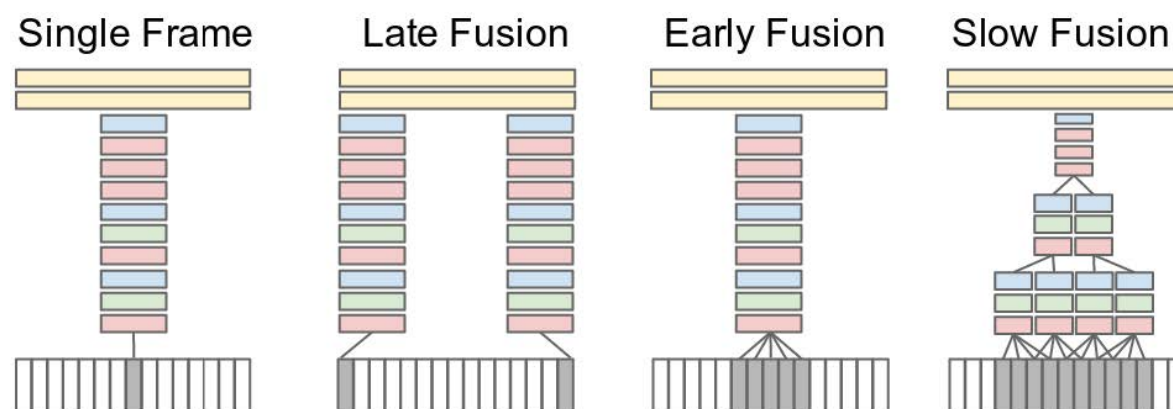
这类方法把视频看作一系列图像的集合，每帧图像单独提取特征，再融合它们的深度特征。

1. "Large-scale video classification with convolutional neural networks"(CVPR, 2014)

贡献：

- (1). 公布了大规模视频数据集Sport-1M。
- (2). 总结使用卷积神经网络提取时序语义的模型。
- (3). 提出一种由人类视网膜启发的视频理解网络。

时序语义融合



从左到右为单融合, 后融合, 前融合, 缓融合

1. 单融合

识别动作靠场景，识别场景依靠物体，在视频中选取一帧作为代表帧，将视频分类任务转化为图片分类任务。最常用的TSN网络在此基础上进行了延伸。

2. 后融合

同一视频前后相隔15帧的两帧输入神经网络中，在全连接分类网络之前进行融合，即前面的卷积部分不进行时序求解，只有最后的全连接层能接收到时序信息，其在高层语义上特征维度较低难以考虑动作细节。

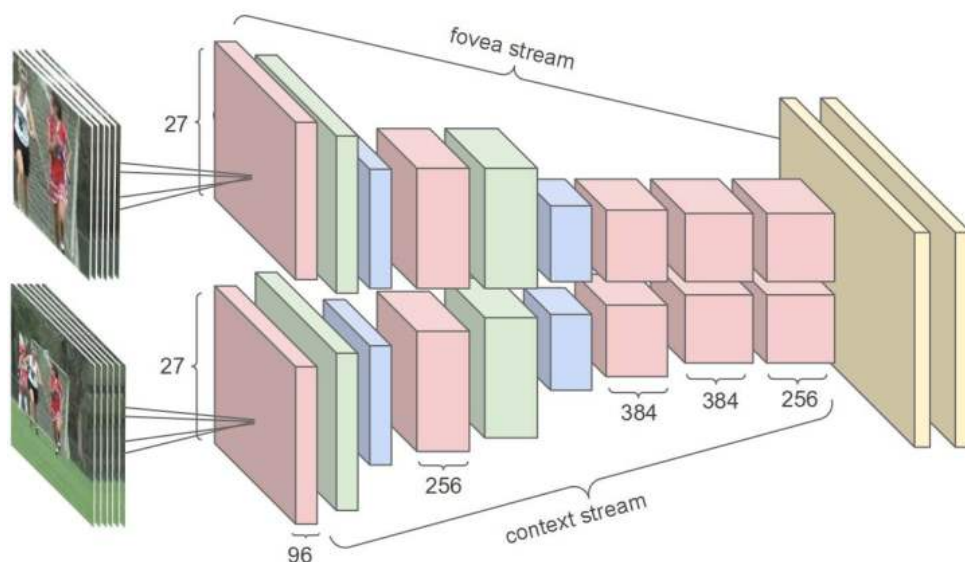
3. 前融合

使用连续的 10 帧进行预测，在第一个卷积层前就将其融合。这里直接将第一层的二维卷积扩展为了深度为 10 (就是 10 帧压成 1 帧) 的三维卷积。在较高分辨率的位置进行融合，卷积层可以更清晰地看到动作的方向和速率，但语义可能较弱。

4. 缓融合

依旧使用连续的 10 帧，将 $[0, 10)$ 帧以 4 为长度，2 为步长，分为 $[0, 4)$ 、 $[2, 6)$ 、 $[4, 8)$ 、 $[6, 10)$ 四段分别进行前融合式卷积。之后在这四部分局部时序语义分为前后两组进行同样的前融合式卷积，最后将这两组再进行融合。总体而言该过程将感受野从 4 帧提至 8 帧最后提至 10 帧(效果最好)。

多分辨率网络



该网络基于上述的任一时序融合模型, 以 178×178 分辨率帧为原始数据. 其中一支接收 178×178 分辨率全尺寸下采样缩放为 89×89 (即缩放为之前的四分之一) 的帧, 即上下文网络. 另一支接收同样分辨率中央裁剪为 89×89 的帧 (即不进行缩放变换), 即中央凹网络.** 由于接收的都是分辨率为 89×89 的帧, 作者去掉了最后一个池化层以维持最终 $7 \times 7 \times 256$ 的特征维度. 在进入全连接分类层前将两支在通道维度上进行连接, 以聚合不同分辨率的信息.

2. "ECO: Efficient Convolutional network for Online video understanding"(ECCV,2018)

Motivation

这篇文章在introduction部分主要提出了两点motivation:

1. 使用单帧的图像, 在很多情况下已经可以获得一个不错的初始分类结果了, 而相邻帧间的很多信息都是冗余的. 因此, ECO中在一个时序邻域内仅使用单帧图像.
2. 为了获得长时程的图像帧间的上下文关系, 仅仅使用简单的分数融合(aggregation)是不够的. 因此, ECO中对较远帧之间采取对feature map 进行3D 卷积的方式进行end-to-end的融合.

相关工作

1. 大部分2-stream类方法和所有3D卷积类型方法都是在学习短时程的时序信息, 即输入一个连续的video clip, 输出一个分类结果. 在video-level上, 通常是对video中选取多个clip分别得到分类结果, 再进行平均得到最后的分类结果, 这样会带来较大的计算开销.
2. 一些方法 (比如去年deeppmind的i3d) 选择增大输入clip的时序长度来获得更长时程的信息, 但这样一方面会带来计算开销的提高, 另外一方面则还是不能处理较长的视频.
3. 一些方法采用编码方法来获得video-level的表示, 但作者表示这样忽略了帧间的信息.
4. 与ECO最相似的是目前被广泛使用的TSN网络:
 - ECO和TSN的相似点: 两者都是从video中均匀采样固定数量的视频帧, 从而来覆盖长时程的时序结构. 两者都是end-to-end训练的.
 - ECO和TSN的不同点: TSN中直接用late-fusion的方式对每个segment的结果做融合, 而ECO中则是进行了feature map层面的融合 (文中除了3D卷积, 也采用了2D卷积). 此外ECO的速度要远快于TSN.

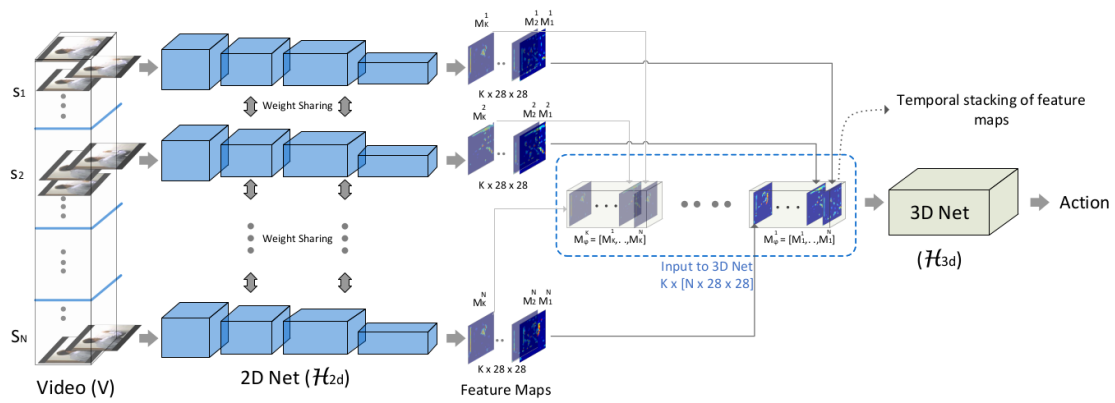


Fig. 1: **Architecture overview of ECO Lite.** Each video is split into N sub-sections of equal size. From each subsection a single frames is randomly sampled. The samples are processed by a regular 2D convolutional network to yield a representation for each sampled frame. These representations are stacked and fed into a 3D convolutional network, which classifies the action, taking into account the temporal relationship.

ECO网络的基本结构如上图所示， $S1-SN$ 是从视频中采样得到的 N 个RGB的segment。

1. 对于每个segment，采用共享的2D卷积子网络来得到96个 $(28, 28)$ 大小的feature map，堆叠后得到一个 $(N, 28, 28, 96)$ 大小的特征volume。此处子网络使用的是BN-Inception网络中的第一部分（到inception-3c层前）。
2. 对于得到的特征volume，采用一个3D子网络进行处理，直接输出对应动作类别数目的一维向量，此处3D子网络采用3D-Resnet18中的部分层。

如上的两部分，就构建了这篇文章中构建的第一种网络结构ECO-Lite。除了用3D卷积进行融合，还可以同时使用2D卷积，如下图所示，即为ECO-Full网络结构。此处多的一个2D网络分支2D-Nets采用的是BN-Inception网络中inception-4a到最后一个pooling层间的部分，最后再采用average-pooling得到video-level的表示，与3D net的结果concat后再得到最后的action分类结果。

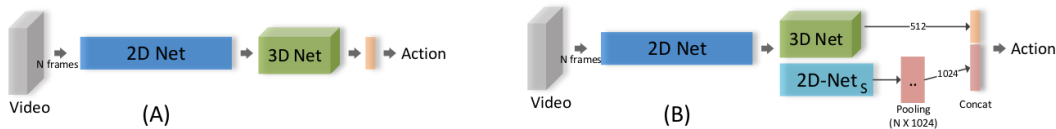


Fig. 2: **(A)** ECO Lite architecture as shown in more detail in Fig. 1. **(B)** Full ECO architecture with a parallel 2D and 3D stream.

采样策略

介绍完网络结构后，再回过头来介绍一下视频帧的采样方法。在ECO中，作者先将一个视频等分为 N 份，再在每份中随机选取一帧作为输入。作者认为这样的随机采样策略可以在训练中引入更多的多样性，并提高泛化能力。

Space-Time Network、3D-CNN

1. "3D Convolutional Neural Networks for Human Action Recognition" (TPAMI 2013)

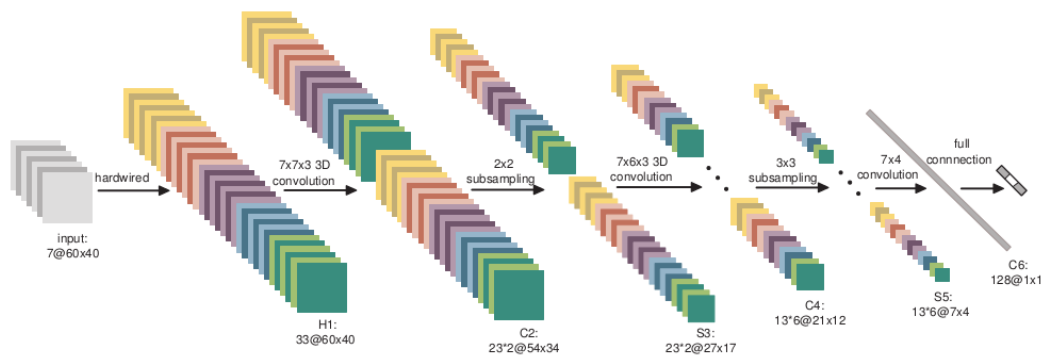
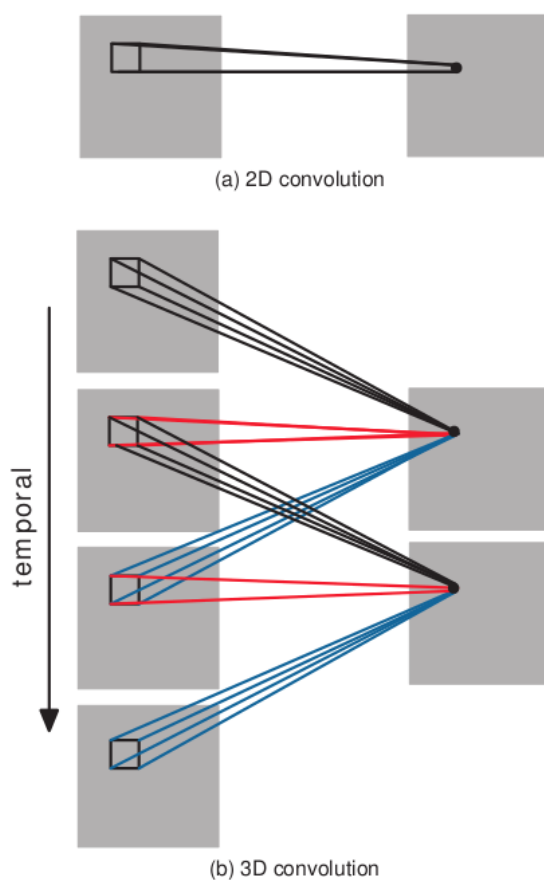


Figure 3. A 3D CNN architecture for human action recognition. This architecture consists of 1 hardwired layer, 3 convolution layers, 2 subsampling layers, and 1 full connection layer. Detailed descriptions are given in the text.

利用CNN解决行为识别的开山之作，引入时间维度（连续帧）使神经网络具有动作识别功能，提取时间与空间特征。初始使用5种*hardwired kernel*: *gray*、*gradient-x*、*gradient-y*、*optflow-x*、*optflow-y*产生33个特征图，使用3D CNN重复conv、pooling操作，最后生成128维全连接层进行分类。其中3D卷积针对三个连续帧运算，具体流程如下图所示：

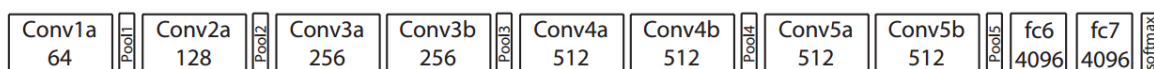


Pros: 在2D CNN的基础上很自然拓展出3D CNN。

Cons: 仅仅考虑短时间信息，不能获取长时间信息。

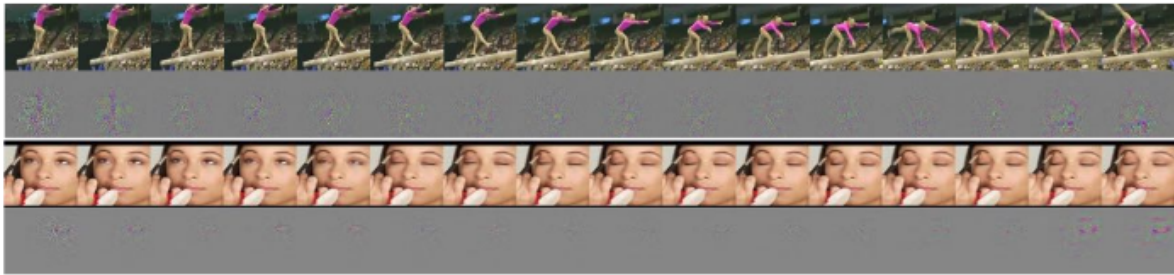
2. "Learning spatiotemporal features with 3d convolutional networks" (ICCV 2015)

该paper提出C3D卷积网络，受到GPU显存的限制，本文仅仅设计出了C3D的网络结构与C3D算子。该网络包含8个卷积层，5个池化层，2个FC、1个softmax。文章中指出不过早池化可以在早期阶段保留更多的时间信息。此外，C3D的视频长度 为16 frame，视频帧尺寸为112*112。



3D CNN反卷积可视化

C3D第五层conv产生的feature map反卷积可视化实验表明视频前几帧C3D关注视频帧的显著性区域，其后几帧会跟踪显著变化区域。

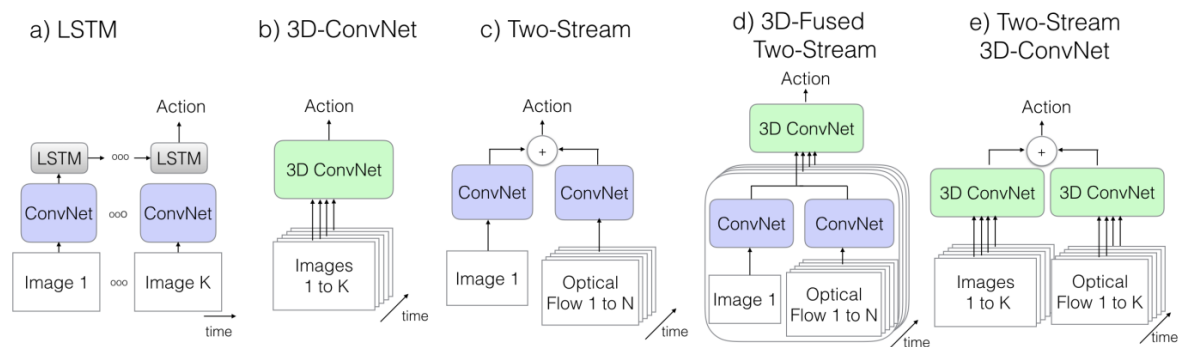


C3D描述子

所谓的C3D描述子，就是C3D网络第一个全连接层输出的特征经过L2归一化后的结果，文章对该描述子的做了实验，表明该描述子有良好的压缩性和泛化性。

3."Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset"(CVPR, 2017)

Github: <https://github.com/MRzzm/action-recognition-models-pytorch> or <https://github.com/deepmind/kinetics-i3d>



上图(e)就是I3D网络，之前的3D ConvNets由于参数比较多，训练数据少，故而网络层数不深。本文考虑在已有图像分类任务的2D ConvNets基础上，想办法将其扩充成3D，然后套入到two-stream方法框架中。因为这些网络本身比较深，又有预先训练好的参数可以用来初始化，所以可以解决参数多、训练数据少的问题。

方法的优越性:

- 通过把2D ConvNet扩充成3D的ConvNet可以在参数初始化的时候用原来在ImageNet上预训练的参数做初始化；
- 还用了比较好的two-stream方法

inflating 2D ConvNets into 3D:

- 直接将那些在图像分类上表现优异的模型扩充成3D模型；
- 在现有的2D architecture基础上将里面的2维卷积核的过滤器filters以及pooling kernels，通常为square:(N,N)，扩充成cubic:*(N,N,N)

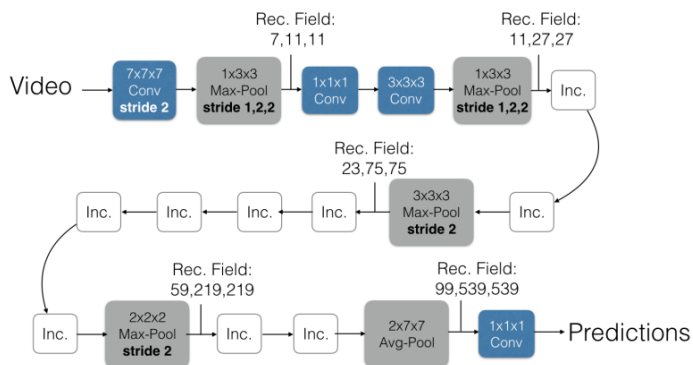
Bootstrapping 3D filters from 2D filters:

直接扩充容易，如何使用原来训练好的模型参数来做初始化。将原来(N,N)的卷积核沿着t的方向复制N遍，得到(N,N,N)，再将这(N,N,N)的卷积核中每个值都除以N(rescaling)就可以满足这个需求。

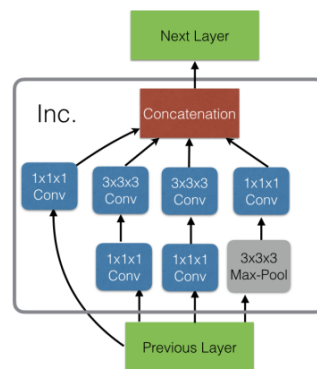
Pacing receptive field growth in space,time and network depth:

扩充后的Inception-V1模型结构:

Inflated Inception-V1



Inception Module (Inc.)

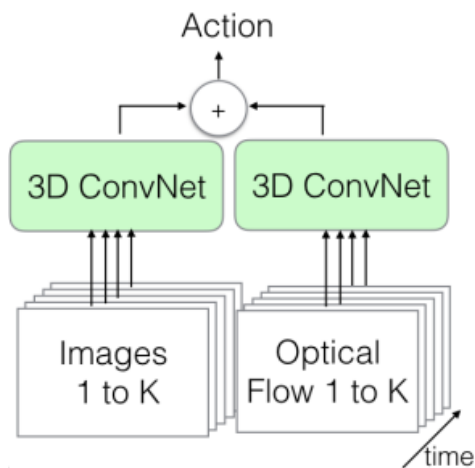


- 可以看到是原来的*Inception-v1*结构变换而来的，原来的*convolution*以及*pooling*全部都成了3D的；
- 大多数保持了对称的特征，例如第一个(7,7)变成(7,7,7)，stride也由原来的(2,2)变成了(2,2,2)
- 但是少数做了改变，比如前面的两个max-pool，并不是(2,2,2)，而是(1,2,2)这样能够更好的保留时间信息；还有最后的这个avg-pool，并不是(7,7,7)而是(2,7,7)。

Two 3D Streams:

将上述的3D ConvNet设计成two-stream的方式，具体的结构图如下所示，对左右两个网络分开训练，然后最后对它们各自的预测结果做平均。

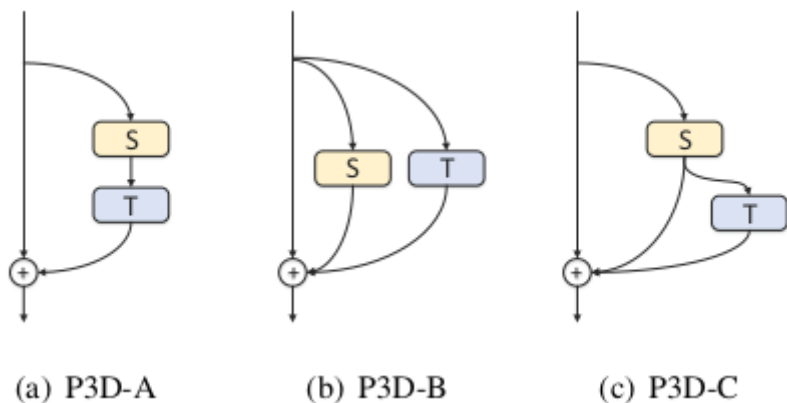
e) Two-Stream 3D-ConvNet



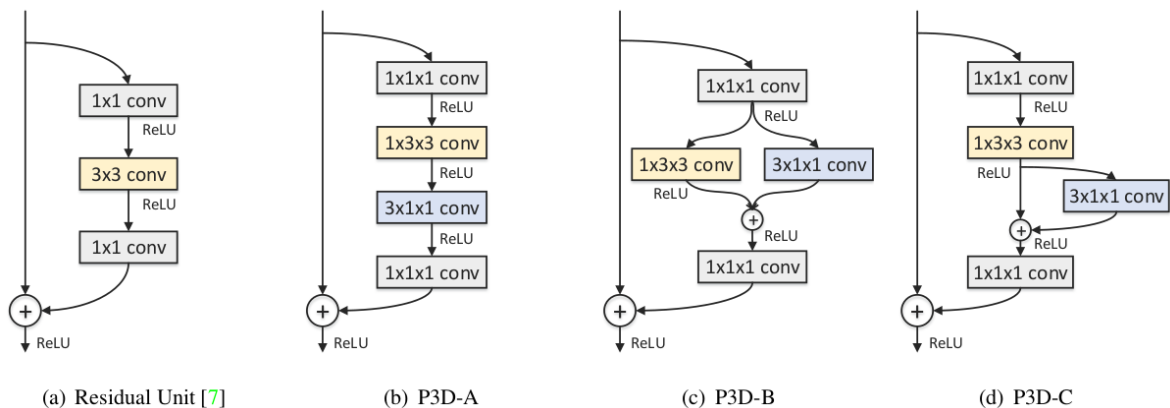
4." Learning spatio-temporal representation with pseudo-3D residual networks."(ICCV, 2017)

Github: <https://github.com/ZhaofanQiu/pseudo-3d-residual-networks>

主要思想为将3D卷积核 $3 \times 3 \times 3$ 解耦为 $1 \times 3 \times 3$ 和 $3 \times 1 \times 1$ 的卷积核，这样不仅能够减少参数数量还能够利用预训练好的2D-CNN，利用从图像中学习的场景和对象的知识使伪3D-CNN性能更好。*P3D Blocks*的灵感来源于ResNet，*P3D Blocks*用于代替ResNet的Residual Units。为了研究空间维度(S)上的2D卷积核和时域(T)上的1D卷积核并联还是串联效果好，设计了三种结构：



具体而言，上图三种结构可以用下图详细描述：



在论文中，作者为了评估上述的三种模块，将其设计到*ResNet-50*中去，分别对其分类性能、时间进行测试。除此之外，选择三个不同的任务：行为识别、动作相似性判断、场景识别，结果显示效果很好，证明了P3D对视频特征提取有良好的效果。

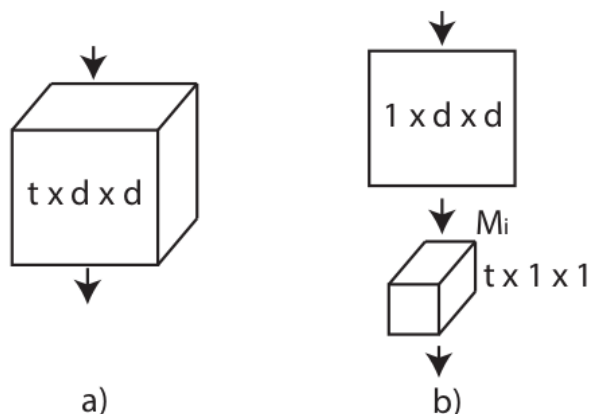
5."A closer look at spatio-temporal convolutions for action recognition."(CVPR, 2018)

(2+1)D依旧尝试用二维卷积和一维卷积来近似表示全三维卷积，即空间与时间建模两个单独的步骤。

与全三维卷积相对比，(2+1)D具有两个优点：

- (2+1)D块中的参数数量与全卷积参数数量相近，虽然没有减少参数的数量，但是依托二维和一维卷积之间的非线性激活函数，网络中非线性得到了进一步提升，增加了可表示函数的复杂性。
- 通过三维卷积的分割，时间与空间成分分割，使得优化更加容易，即相同容量的三维卷积相比具有训练误差更低的优势。

其中，(2+1)D的结构如下所示：



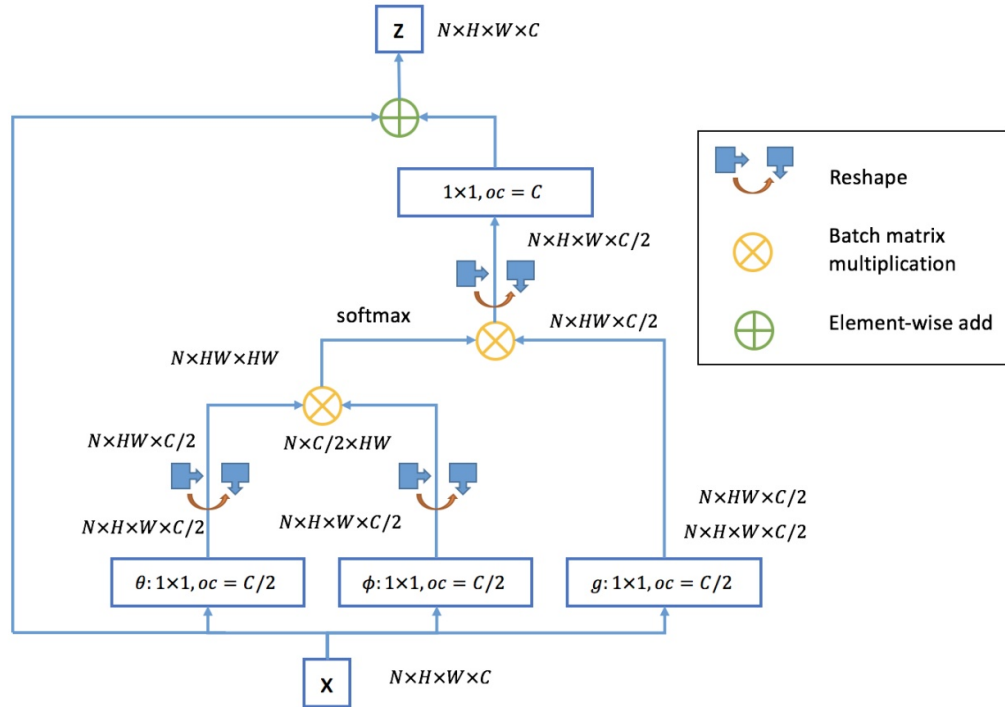
6. "Non-local neural networks"(CVPR, 2018)

Github: https://github.com/AlexHex7/Non-local_pytorch

其网络框架如下图所示，3D卷积感受野是有限区域，而NL旨在解决长距离依赖问题，NL的响应是所有空间和时间位置特征的加权平均，其中为了使得：

$$\mathbf{y}_i := \frac{1}{C(\mathbf{x}_i)} \sum_j f(\mathbf{x}_i, \mathbf{x}_j) \mathbf{g}(\mathbf{x}_j)$$

其中, $f(\mathbf{x}_i, \mathbf{x}_j)$ 用于度量相似性, $\mathbf{g}(\mathbf{x}_j)$ 计算响应, $C(\mathbf{x}_i)$ 用于归一化。当 $f(\mathbf{x}_i, \mathbf{x}_j) = w_{ij}, \mathbf{g}(\mathbf{x}_j) = \mathbf{x}_j$ 时, NL操作退化为全连接层；当 $f(\mathbf{x}_i, \mathbf{x}_j) = \exp \mathbf{x}_i^\top \mathbf{x}_j$ 时, NL操作 $\mathbf{y}_i = \text{softmax}(\mathbf{x}_i^\top \mathbf{x}_j) \mathbf{g}(\mathbf{x}_j)$ 退化为self-attention。实验中发现non-local block加在底层比加在高层效果要好，加多个non-local blocks会有效果提升但不是很明显。



7. "SlowFast Networks for Video Recognition" (CVPR, 2019, FAIR)

Github: <https://github.com/facebookresearch/SlowFast>

SlowFast使用了一个慢速高分辨率CNN（Slow通道）来分析视频中的静态内容，同时使用一个快速低分辨率CNN（Fast通道）来分析视频中的动态内容。这一技术部分源于灵长类动物的视网膜神经节的启发，视网膜神经节中，大约80%的细胞（P-cells）以低频运作，可以识别细节，而大约20%的细胞（M-cells）则以高频运作，负责响应快速变化。类似的，在SlowFast中，Slow通道的计算成本要比Fast通道高4倍。

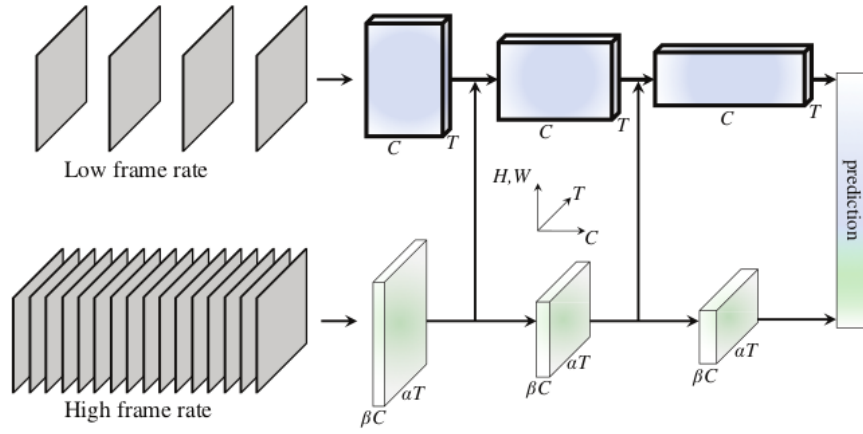


Figure 1. A **SlowFast network** has a low frame rate, low temporal resolution *Slow* pathway and a high frame rate, $\alpha \times$ higher temporal resolution *Fast* pathway. The Fast pathway is lightweight by using a fraction (β , e.g., 1/8) of channels. Lateral connections fuse them.

SlowFast工作原理

*Slow*通道和*Fast*通道都使用3D-ResNet模型，捕捉若干帧之后立即运行3D卷积操作。

*Slow*通道使用一个较大的时序跨度（即每秒跳过的帧数），通常设置为16，这意味着大约1秒可以采集2帧。*Fast*通道使用一个非常小的时序跨度 t/a ，其中 a 通常设置为8，以便1秒可以采集15帧。*Fast*通道通过使用小得多的卷积宽度（使用的滤波器数量）来保持轻量化，通常设置为慢通道卷积宽度的 $1/\alpha$ ，这个值被标记为 β 。使用小一些的卷积宽度的原因是*Fast*通道需要的计算量要比*Slow*通道小4倍，虽然它的时序频率更高。

stage	<i>Slow</i> pathway	<i>Fast</i> pathway	output sizes $T \times S^2$
raw clip	-	-	64×224^2
data layer	stride 16, 1^2	stride 2, 1^2	<i>Slow</i> : 4×224^2 <i>Fast</i> : 32×224^2
conv ₁	1×7^2 , 64 stride 1, 2^2	5×7^2 , 8 stride 1, 2^2	<i>Slow</i> : 4×112^2 <i>Fast</i> : 32×112^2
pool ₁	1×3^2 max stride 1, 2^2	1×3^2 max stride 1, 2^2	<i>Slow</i> : 4×56^2 <i>Fast</i> : 32×56^2
res ₂	$\begin{bmatrix} 1 \times 1^2, 64 \\ 1 \times 3^2, 64 \\ 1 \times 1^2, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 3 \times 1^2, 8 \\ 1 \times 3^2, 8 \\ 1 \times 1^2, 32 \end{bmatrix} \times 3$	<i>Slow</i> : 4×56^2 <i>Fast</i> : 32×56^2
res ₃	$\begin{bmatrix} 1 \times 1^2, 128 \\ 1 \times 3^2, 128 \\ 1 \times 1^2, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 3 \times 1^2, 16 \\ 1 \times 3^2, 16 \\ 1 \times 1^2, 64 \end{bmatrix} \times 4$	<i>Slow</i> : 4×28^2 <i>Fast</i> : 32×28^2
res ₄	$\begin{bmatrix} 3 \times 1^2, 256 \\ 1 \times 3^2, 256 \\ 1 \times 1^2, 1024 \end{bmatrix} \times 6$	$\begin{bmatrix} 3 \times 1^2, 32 \\ 1 \times 3^2, 32 \\ 1 \times 1^2, 128 \end{bmatrix} \times 6$	<i>Slow</i> : 4×14^2 <i>Fast</i> : 32×14^2
res ₅	$\begin{bmatrix} 3 \times 1^2, 512 \\ 1 \times 3^2, 512 \\ 1 \times 1^2, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 3 \times 1^2, 64 \\ 1 \times 3^2, 64 \\ 1 \times 1^2, 256 \end{bmatrix} \times 3$	<i>Slow</i> : 4×7^2 <i>Fast</i> : 32×7^2
global average pool, concat, fc			# classes

上图是一个SlowFast网络的实例。卷积核的尺寸记作 $\{T \times S^2, C\}$ ，其中 T 、 S 和 C 分别表示时序temporal, 空间spatial和频道Channel的尺寸。跨度记作 $\{temporal\ stride, spatial\ stride^2\}$ 。速度比率(跳帧率)为 $\alpha = 8$ ，频道比率为 $1/\beta = 1/8$ ， τ 设置为16。绿色表示高一些的时序分辨率，Fast通道中的橙色表示较少的频道。

侧向连接

如图中所示，来自Fast通道的数据通过侧向连接被送入Slow通道，这使得Slow通道可以了解Fast通道的处理结果。单一数据样本的形状在两个通道间是不同的（Fast通道是 $\{\alpha T, S^2, \beta C\}$ ，而Slow通道是 $\{T, S^2, \alpha \beta C\}$ ），这要求SlowFast对Fast通道的结果进行数据变换，然后融入Slow通道。论文给出了三种进行数据变换的技术思路，其中第三个思路在实践中最有效。

- *Time-to-channel*: 将 $\{\alpha T, S^2, \beta C\}$ 变形转置为 $\{T, S^2, \alpha \beta C\}$ ，就是说把 α 帧压入一帧
- *Time-strided*采样: 简单地每隔 α 帧进行采样， $\{\alpha T, S^2, \beta C\}$ 就变换为 $\{T, S^2, \beta C\}$
- *Time-strided*卷积: 用一个 5×12 的核进行3D卷积， $2\beta C$ 输出频道，跨度 $= \alpha$ 。

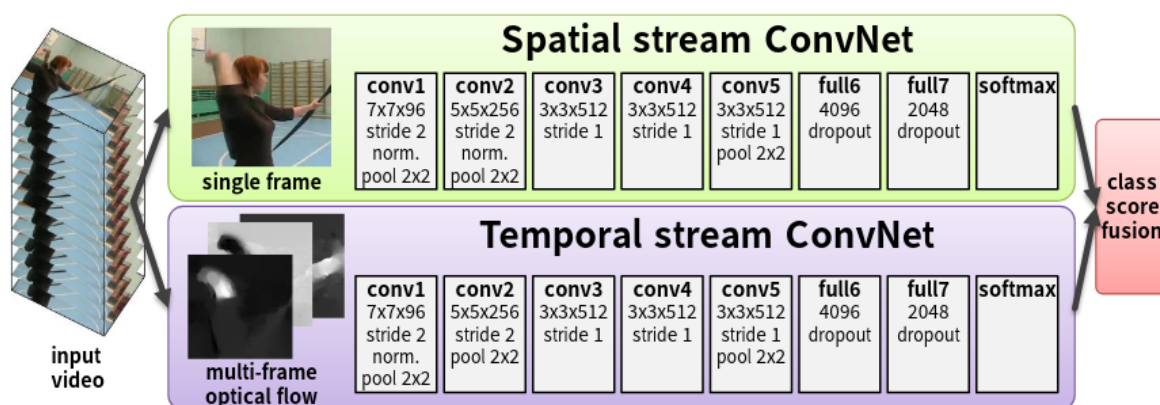
研究人员发现双向侧链接，即将Slow通道结果也送入Fast通道，对性能没有改善。

在每个通道的末端，SlowFast执行全局平均池化，一个用来降维的标准操作，然后组合两个通道的结果并送入一个全连接分类层，该层使用softmax来识别图像中发生的动作。

Multi-Stream Network

1. "two-stream convolutional networks for action recognition in videos," (NIPS, 2014)

由于视频天生自带空间与时间内容，该文提出了经典的双流网络，该网络由两个子网络组成：*spatial stream convnet* 和 *temporal stream convnet*，空间流卷积网络指每一帧的表面信息，光流卷积网络则代表帧与帧之间的运动信息。每一个子网络均由CNN与softmax得分组成，最后时间与空间得到的softmax通过两种融合策略给出分类：平均、在堆叠softmax上训练SVM，具体的网络框架如下图所示：



光流卷积网络考虑四种输入方式：光流栈、轨迹叠加、双向光流、减去平均光流。

空间卷积网络则输入静态图像，一般采用ImageNet预训练模型。

考虑到当时视频数据集的规模比较小（主要指的是UCF-101和HMDB-51这两个数据集，训练集数量分别是9.5K和3.7K个video），因此作者采用了*multi-task*策略解决上述问题，即光流卷积网络最后的softmax层拓展成两层，一层计算HMDB-51数据集分类输出，一层计算UCF-101数据集分类输出，所以存在两个task与两个loss值，最终回传loss设为两条支路loss sum。

Pros: 可以使用预训练好的2D卷积网络。

Cons: 子网络之间的interaction非常困难，对学习时空特征影响很大。

2. "Temporal segment networks: Towards good practices for deep action recognition." (ECCV, 2016)

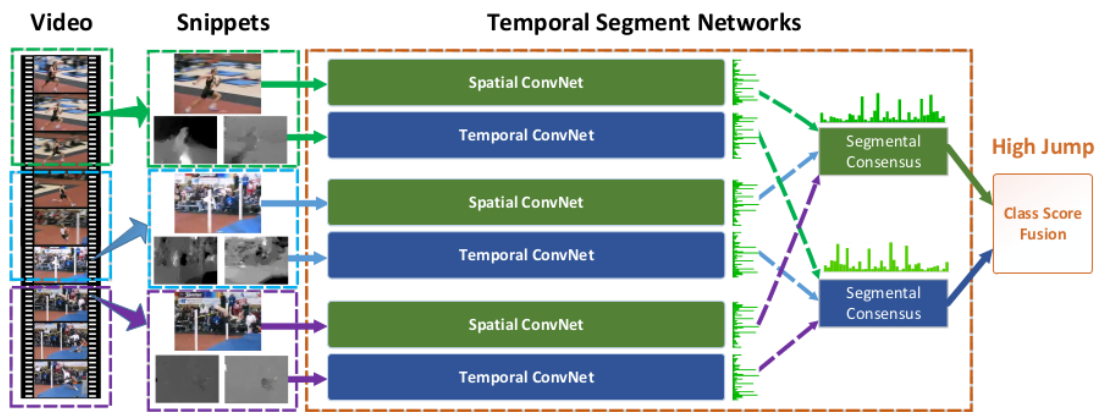


Fig. 1. Temporal segment network: One input video is divided into K segments and a short snippet is randomly selected from each segment. The class scores of different snippets are fused by an the segmental consensus function to yield segmental consensus, which is a video-level prediction. Predictions from all modalities are then fused to produce the final prediction. ConvNets on all snippets share parameters.

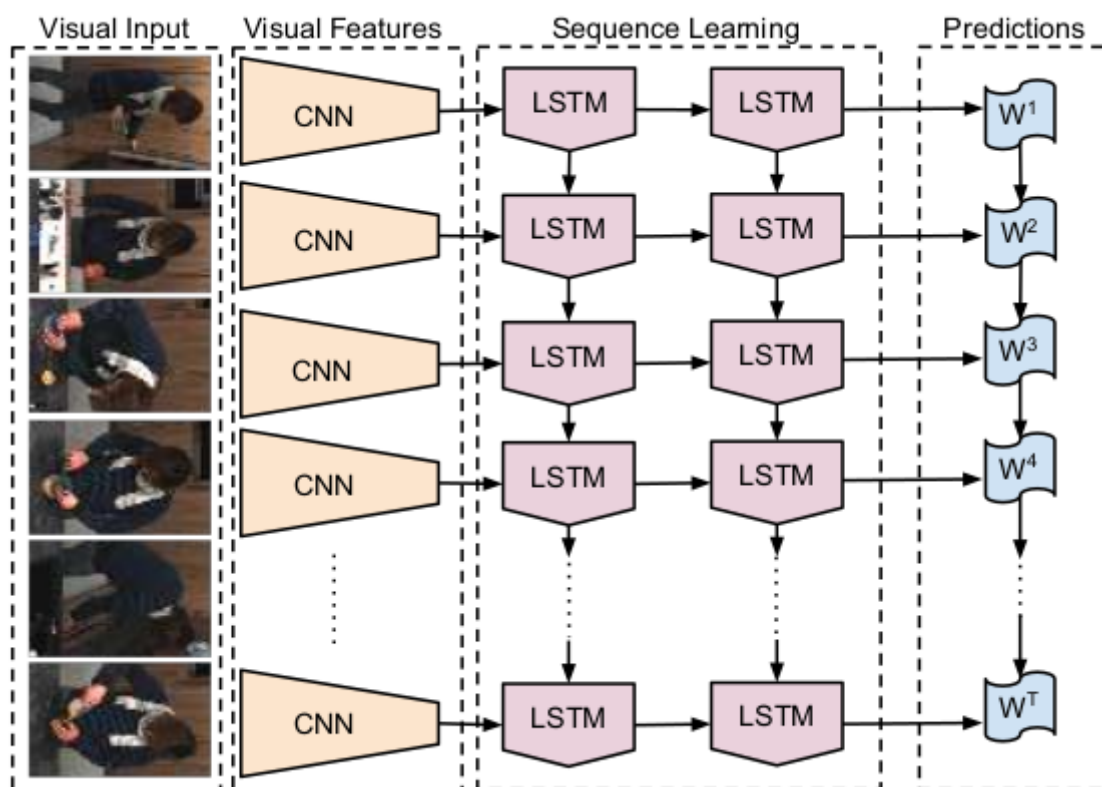
网络主体和two-stream算法一样分为spatial stream convnet(上图绿色方块) 和 temporal stream convnet(上图蓝色方块), 只不过使用了更深的BN-Inception网络, 最后融合了多个采样识别结果。

训练样本随机采样, 即把视频平均分为 K 份, 图中为3份, 每份中随机取1帧RGB图像作为spatial convnetde 输入, 一定数量光流作为temporal convnet输入, 每segment可以得到一个分类分数, 图中spatial和temporal各有3个分数, 再将这些分数使用某种方法(方法有: evenly averaging, maximum, weighted averaging)进行融合得到各自类别分数, 在训练中spatial和temporal是分开训练的, 所以可以单独使用, 同时使用spatial和temporal预测时, 需要加一个权重。

Hybrid Network

一种获取时间信息的做法是在CNN上面增加循环层(recurrent layer)如LSTM, 从而构建出一种混合型网络(Hybrid Network), 该类型网络可以兼顾CNN与LSTM的优势。

1. "Long-term Recurrent Convolutional Networks for Visual Recognition and Description" (CVPR, 2015)



这篇文章中提出的LRCN结构如上图所示，对于视频图片帧，LRCN先利用CNN提出输入图片帧的特征信息并将这些具备时序相关的图片特征输入到后续LSTM网络中处理，进而得到一个时序的输出。LRCN得益于LSTM特性，输入可以是单帧图片、视频帧，网络中的数据也可以是单个预测之或者序列预测值，进而LRCN能够适应多种任务处理:行为识别、图片描述、视频描述(CRF替换CNN)。

Pros:借助现有的模型，便于搭建网络框架。

Cons:微调难度大。

2."RPAN: An end-to-end recurrent pose-attention network for action recognition in videos" (ICCV Oral, 2017)

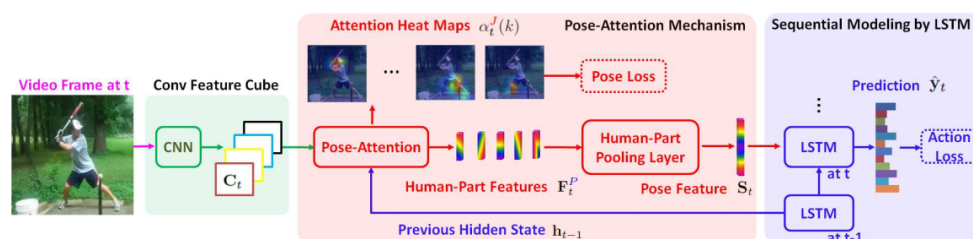


Figure 2. Our End-to-End Recurrent Pose-Attention Network (RPAN). At the t -th step, the video frame is fed into CNN to generate the convolutional feature cube C_t . Then, with guidance of the previous hidden state h_{t-1} of LSTM, our pose attention mechanism learns several human-part-related features F_t^P from C_t . As attention parameters are partially shared on the semantic-related human joints belonging to the same body part, our human-part-related features encode robust body-structure-information to discriminate complex actions. Finally, these features are fed into the human-part pooling layer to produce a highly-discriminative pose-related feature S_t , which is the input to LSTM for action recognition. The whole RPAN can be efficiently trained in an end-to-end fashion, by considering the action loss (prediction \hat{y}_t vs. action label) and the pose loss (attention heat maps $\alpha_t^j(k)$ vs. pose annotations) together.

整个网络框架可以分成三个大的部分：

- 特征生成部分：Conv Feature cube from CNN
- 姿态注意机制：Pose-Attention Mechanism
- LSTM：RNN网

行为/动作预测

Action Prediction
























































行为预测分为 *short-term prediction*、*long-term prediction*，其中 *short-term prediction* 主要处理短视频(通常只有几秒)，此时 *short-term prediction* 的目标是希望根据时序上不完整的视频推断动作的标签。*long-term prediction* 则是基于观察到的动作与行为推断出未来会发生的动作，这两个动作是独立的、时间相关的。

<https://zhuanlan.zhihu.com/p/86184886>

<https://zhuanlan.zhihu.com/p/86185203>

Motion Trajectory Prediction

PapersWithCode/Trajectory Prediction/Leaderboards:

	ETH/UCY	 Transformer TF	Transformer Networks for Trajectory Forecasting			See all
	ActEV	 Next	Peeking into the Future: Predicting Future Person Activities and Locations in Videos			See all
	Stanford Drone	 DAG-Net	DAG-Net: Double Attentive Graph Neural Network for Trajectory Forecasting			See all
	GPS	 Support Vector Machines	Inferring hybrid transportation modes from sparse GPS data using a moving window SVM classification			See all
	ETH BIWI Walking Pedestrians dataset	 Social Ways	Social Ways: Learning Multi-Modal Distributions of Pedestrian Trajectories with GANs			See all
	Hotel BIWI Walking Pedestrians dataset	 Social Ways	Social Ways: Learning Multi-Modal Distributions of Pedestrian Trajectories with GANs			See all
	TRAF	 TraPHic	TraPHic: Trajectory Prediction in Dense and Heterogeneous Traffic Using Weighted Interactions			See all
	NGSIM	 TraPHic	TraPHic: Trajectory Prediction in Dense and Heterogeneous Traffic Using Weighted Interactions			See all
	Argoverse	 SpectralCows	Forecasting Trajectory and Behavior of Road-Agents Using Spectral Clustering in Graph-LSTMs			See all
	Lyft Level 5	 SpectralCows	Forecasting Trajectory and Behavior of Road-Agents Using Spectral Clustering in Graph-LSTMs			See all
	Apolloscape	 SpectralCows	Forecasting Trajectory and Behavior of Road-Agents Using Spectral Clustering in Graph-LSTMs			See all
	STATS SportVu NBA [ATK]	 DAG-Net	DAG-Net: Double Attentive Graph Neural Network for Trajectory Forecasting			See all
	STATS SportVu NBA [DEF]	 DAG-Net	DAG-Net: Double Attentive Graph Neural Network for Trajectory Forecasting			See all
	ForkingPaths	 Multiverse	The Garden of Forking Paths: Towards Multi-Future Trajectory Prediction			See all

1. "Social GAN: Socially Acceptable Trajectories with Generative Adversarial Networks" (CVPR, 2018)

Github: <https://github.com/agrimgupta92/sgan>

2. "Peeking into the Future: Predicting Future Person Activities and Locations in Videos" (CVPR, 2019)

Github: <https://github.com/google/next-prediction> (Next Model, TensorFlow, Python2.7)

SOTA for Activity Prediction on **ActEV**

3. "StarNet: Pedestrian Trajectory Prediction using Deep Neural Network in Star Topology" (2019)

4. "Social Ways: Learning Multi-Modal Distributions of Pedestrian Trajectories with GANs" (CVPR, 2019)

Github: <https://github.com/amiryanj/socialways> (Python3, PyTorch)

SOTA for Trajectory Prediction on **ETH BIWI Walking Pedestrians dataset**

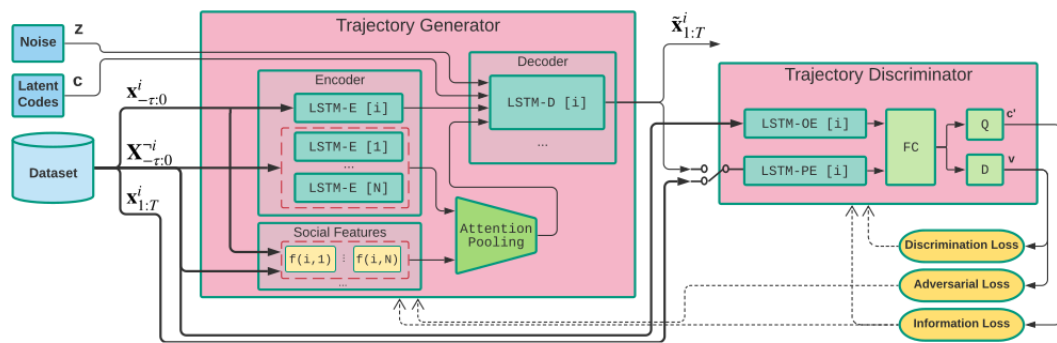


Figure 2. Block Diagram of the Social Ways prediction system. The yellow ellipses represent loss calculations. The dashed arrows show the backpropagation directions. The bold arrows carry ground truth data.

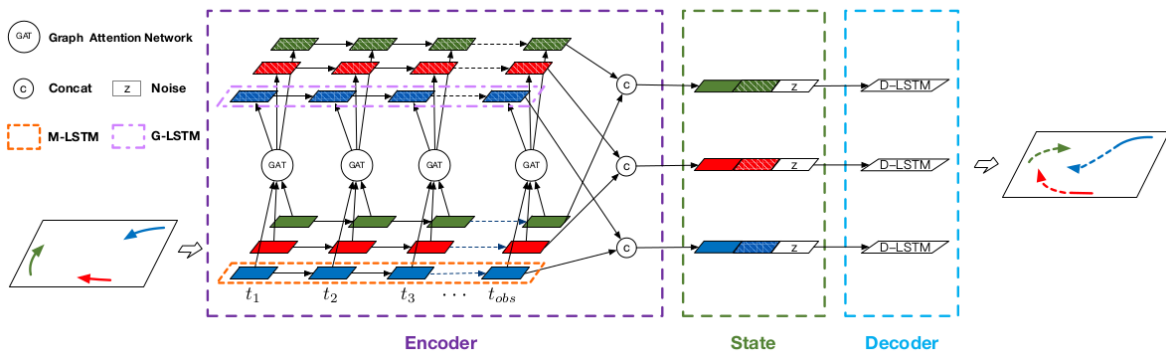
如上图所示，文章的基本框架是GAN网络，在不考虑batch批处理的情况下，模型逐一为每个行人预测轨迹。

- 在Generator中，对于待预测行人 i ，首先会将所有行人的已知轨迹进行编码，而后基于 i 和其他行人之间的地理和运动信息，引入注意力机制使得模型对其他行人的交互信息自主适应。行人 i 的轨迹编码、注意力池化后的交互信息、噪音、latent code (c 输入控制生成的分布) 四种输入作为Decoder的输入，解码出行人 i 的预测轨迹。
- 在Discriminator中，会对生成轨迹/真实轨迹进行判别，判别的结果作为Generator/Discriminator的代价函数。
- 模型框架具体来说是InfoGAN，InfoGAN网络解决的是在无监督的情况下通过修改latent code 倾向从而控制GAN的生成分布，与GAN相比其强调latent code对生成的控制性，与cGAN相比其强调能够在有潜在类别的数据中无监督（无数据标签）学习。因而GAN网络中新引入了Latent Code和Information Loss两个结构。

参考: <https://www.cnblogs.com/sinoyou/p/11512830.html>

5. "STGAT: Modeling Spatial-Temporal Interactions for Human Trajectory Prediction" (ICCV, 2019)

Github: <https://github.com/huang-xx/STGAT> (Python3, PyTorch)



简述: *STGAT*基于*Seq2Seq*结构, 重点在于利用*GAT* (Graph Attention Network) 提取交互信息, 属于轻量级网络, *STGAT*同时还加入了固定先验分布的噪声以生成多样性的轨迹。

Highlights:

- 序列*GAT*网络: 利用序列模型编码每一条轨迹序列, 每一步都存在序列模型都会生成状态, 而后针对每个个体, 使用*GAT*网络计算注意力权重, 将这些状态加权平均, 形成该个体在时刻的交互信息。
- 在neck vector处新增了噪音, 但是由于缺少了*GAN*、*VAE*等生成模型的结构, 不太清楚模型对噪音-行为模式关联度的学习效果。实验中的multi modal评价的图比较少。
- 模型可解释性: 利用*GAT*中的注意力机制, 在绘制预测轨迹时显示了在不同时刻的交互信息的重要性。

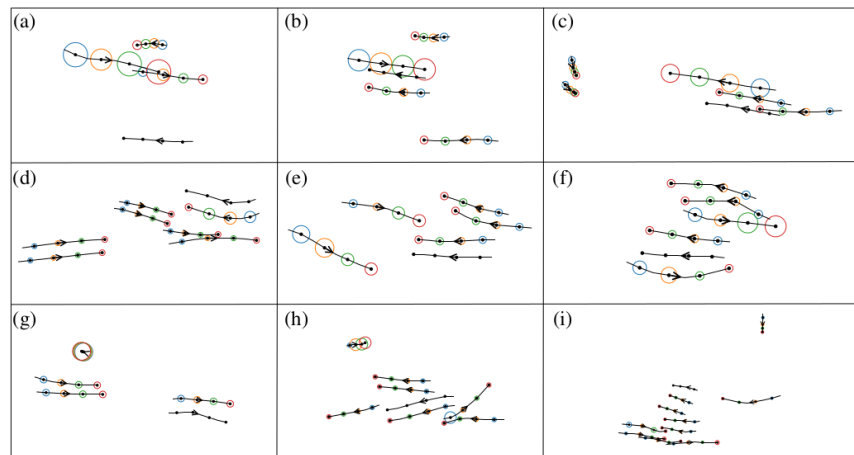


Figure 8. Attention weights predicted by graph attention mechanism. The solid dots on the trajectory indicate different time-steps and the arrows show the directions of trajectories. The trajectories without circles are the target pedestrians. The circles on other trajectories show the attentions represented with the radius proportional to attention weight.

6."DAG-Net: Double Attentive Graph Neural Network for Trajectory Forecasting"(2020)

Github:<https://github.com/alexmonti19/dagnet> (Python3.8, PyTorch1.5, CUDA10.0)

SOTA for Trajectory Prediction on **STATS SportVu NBA [ATK / DEF]** , **Stanford Drone**

7."The Garden of Forking Paths: Towards Multi-Future Trajectory Prediction"(CVPR, 2020)

Github:<https://github.com/JunweiLiang/Multiverse> (**Multiverse** Model, Python2/3, TensorFlow-GPU-1.15.0)

SOTA for Trajectory Prediction on **ForkingPaths**

1.提出了新的数据集： *The Forking Paths Dataset*

2.提出新模型： *Multiverse*

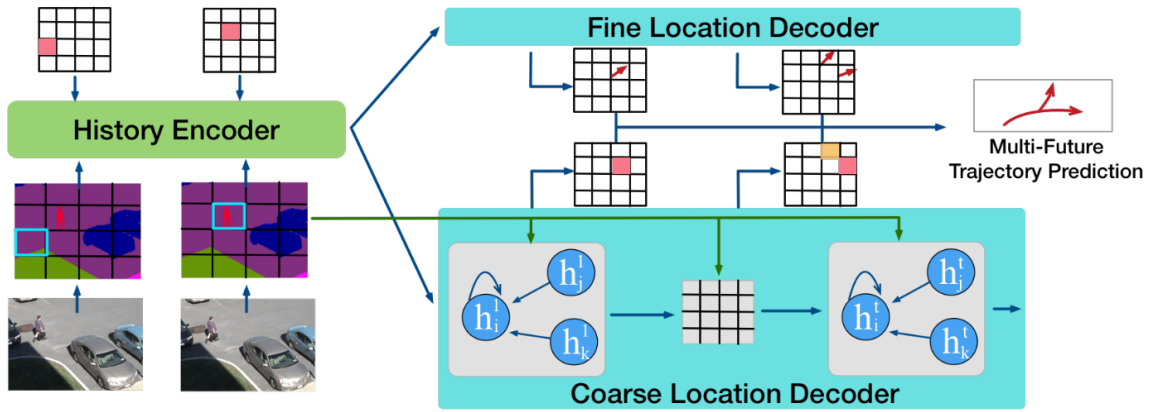


Figure 2: Overview of our model. The input to the model is the ground truth location history, and a set of video frames, which are preprocessed by a semantic segmentation model. This is encoded by the “History Encoder” convolutional RNN. The output of the encoder is fed to the convolutional RNN decoder for location prediction. The coarse location decoder outputs a heatmap over the 2D grid of size $H \times W$. The fine location decoder outputs a vector offset within each grid cell. These are combined to generate a multimodal distribution over \mathbb{R}^2 for predicted locations.