Deep Learning In Video Recognition

Yogesh Bhardwaj
Dept of Computer Applications
Invertis University, Bareilly U. P

I. INTRODUCTION

Medical images like MRIs, CTs (3D images) are very similar to videos - both of them encode 2D spatial information over a 3rd dimension. Much like diagnosing abnormalities from 3D images, action recognition from videos would require capturing context from entire video rather than just capturing information from each frame.

In this post, I summarize the literature on action recognition from videos. The post is organized into three sections -

A. What is action recognition and why is it tough
B. Overview of approaches
C. Summary of papers

A. Action recognition and why is it tough?

Action recognition task involves the identification of different actions from video clips (a sequence of 2D frames) where the action may or may not be performed throughout the entire duration of the video. This seems like a natural extension of image classification tasks to multiple frames and then aggregating the predictions from each frame. Despite the stratospheric success of deep learning architectures in image classification (ImageNet), progress in architectures for video classification and representation learning has been slower.

What made this task tough?

1. Huge Computational Cost

A simple convolution 2D net for classifying 101 classes has just ~5M parameters whereas the same architecture when inflated to a 3D structure results in ~33M parameters. It takes 3 to 4 days to train a 3DConvNet on UCF101 and about two months on Sports-1M, which makes extensive architecture search difficult and overfitting likely[1].

2. Capturing long context

Action recognition involves capturing spatiotemporal context across frames. Additionally, the spatial information captured has to be compensated for camera movement. Even having strong spatial object detection doesn't suffice as the motion information also carries finer details. There's a local as well as global context w.r.t. motion information which needs to be captured for robust predictions. For example, consider the video representations shown in Figure 2. A strong image classifier can identify human, water body in both the videos but the nature of temporal periodic action differentiates front crawl from breast stroke.

a. Designing Classification Architecture

Designing architectures that can capture spatiotemporal information involve multiple options which are non-trivial and expensive to evaluate. For example, some possible strategies could be

- One network for capturing spatiotemporal information vs. two separate ones for each spatial and temporal
- Fusing predictions across multiple clips
- End-to-end training vs. feature extraction and classifying separately
- Fusing predictions

b. No standard benchmark

The most popular and benchmark datasets have been UCF101 and Sports1M for a long time. Searching for reasonable architecture on Sports1M can be extremely expensive. For UCF101, although the number of frames is comparable to ImageNet, the high spatial correlation among the videos makes the actual diversity in the training much lesser. Also, given the similar theme (sports) across both the datasets, generalization of benchmarked architectures to other tasks remained a problem. This has been solved lately with the introduction of Kinetics dataset[2].

Fig 1: Sample illustration of UCF-101.

It must be noted here that abnormality detection from 3D medical images doesn't involve all the challenges mentioned here.

The major differences between action recognition from medical images are mentioned as below

i. In case of medical imaging, the temporal context may not be as important as action recognition. For example, detecting hemorrhage in a head CT scan could involve much less temporal context across slices. Intracranial hemorrhage can be detected from a single slice only. As opposed to that, detecting lung nodule from chest CT scans would involve capturing temporal context as the nodule as well as bronchi and vessels all look like circular objects in 2D scans. It's only when 3D context is captured, that nodules can be seen as spherical objects as opposed to cylindrical objects like vessels

ii. In case of action recognition, most of the research ideas resort to using pre-trained 2D CNNs as a starting point for drastically better convergence. In case of medical images, such pre-trained networks would be unavailable.

B. Overview of approaches

Before deep learning came along, most of the traditional CV algorithm variants for action recognition can be broken down into the following 3 broad steps:

1. Local high-dimensional visual features that describe a region of the video are extracted either densely [3] or at a sparse set of interest points[4 , 5].
2. The extracted features get combined into a fixed-sized video level description. One popular variant to the step is to bag of visual words (derived using hierarchical or k-means clustering) for encoding features at video-level.
3. A classifier, like SVM or RF, is trained on bag of visual words for final prediction

Of these algorithms that use shallow hand- crafted features in Step 1, improved Dense Trajectories [6] (iDT) which uses densely sampled trajectory features was the state-of-the- art. Simultaneously, 3D convolutions were used as is for action recognition without much help in 2013[7]. Soon after this in 2014, two breakthrough research papers were released which form the backbone for all the papers we are going to discuss in this post. The major differences between them was the design choice around combining spatiotemporal information.

Approach 1: Single Stream Network

In this work [June 2014], the authors - Karpathy et al. - explore multiple ways to fuse temporal information from consecutive frames using 2D pre-trained convolutions.

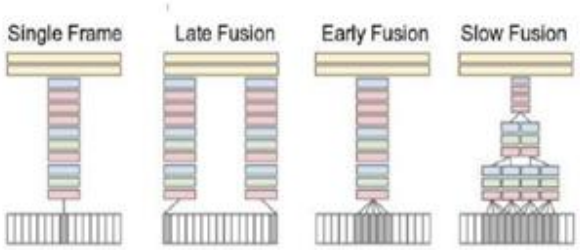

Fig 2: Fusion Ideas

As can be seen in Fig 2, the consecutive frames of the video are presented as input in all setups. Single frame uses single architecture that fuses information from all frames at the last stage. Late fusion uses two nets with shared params, spaced 15 frames apart, and also combines predictions at the end. Early fusion combines in the first layer by convolving over 10 frames. Slow fusion involves fusing at multiple stages, a balance between early and late fusion. For final predictions, multiple clips were sampled from entire video and prediction scores from them were averaged for final prediction.

Despite extensive experimentations the authors found that the results were significantly worse as compared to state-of-the-art hand-crafted feature based algorithms. There were multiple reasons attributed for this failure:

a. The learnt spatiotemporal features didn't capture motion features
b. The dataset being less diverse, learning such detailed features was tough

Approach 2: Two Stream Networks

In this pioneering work [June 2014] by Simmoyan and Zisserman, the authors build on the failures of the previous work by Karpathy et al. Given the toughness of deep architectures to learn motion features, authors explicitly modeled motion features in the form of stacked optical flow vectors. So instead of single network for spatial context, this architecture has two separate networks - one for spatial context (pre-trained), one for motion context. The input to the spatial net is a single frame of the video. Authors experimented with the input to the temporal net and found bi-directional optical flow stacked across for 10 successive frames was performing best. The two streams were trained

separately and combined using SVM. Final prediction was same as previous paper,
i.e. averaging across sampled frames.

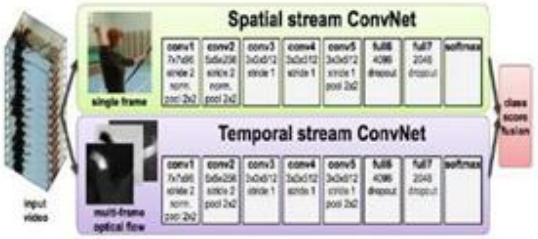

Fig 3: Two stream architecture.

Though this method improved the performance of single stream method by explicitly capturing local temporal movement, there were still a few drawbacks:

a. Because the video level predictions were obtained from averaging predictions over sampled clips, the long range temporal information was still missing in learnt features.
b. Since training clips are sampled uniformly from videos, they suffer from a problem of false label assignemnt. The ground truth of each of these clips are assumed same as ground truth of the video which may not be the case if the action just happens for a small duration within the entire video.
c. The method involved pre-computing optical flow vectors and storing them separately. Also, the training for both the streams was separate implying end-to-end training on- the-go is still a long road.

C. Summaries

Following papers which are, in a way, evolutions from the two papers (single stream and two stream) which are summarized as below:

1. LRCN
2. C3D
3. Conv3D & Attention
4. TwoStreamFusion
5. TSN
6. ActionVlad
7. HiddenTwoStream
8. I3D
9. T3D

The recurrent theme around these papers can be summarized as follows. All of the papers are improvisations on top of these basic ideas.

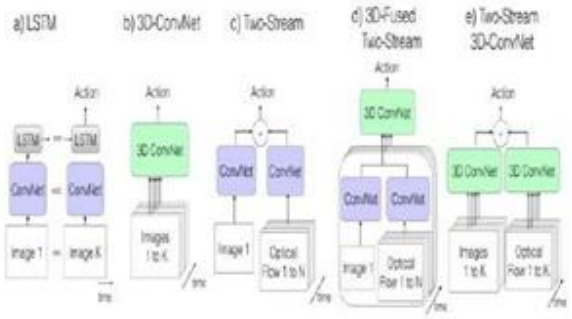

Fig 4: Recurrent theme across papers.

For each of these papers, I list down their key contributions and explain them. I also show their benchmark scores on UCF101- split1.

LRCN

- Long-term Recurrent Convolutional Networks for Visual Recognition and Description
- Donahue et al.
- Submitted on 17 November 2014
- Arxiv Link

Key Contributions:

- Building on previous work by using RNN as opposed to stream based designs
- Extension of encoder-decoder architecture for video representations
- End-to-end trainable architecture proposed for action recognition

Explanation:

In a previous work by Ng et al[9]. authors had explored the idea of using LSTMs on separately trained feature maps to see if it can capture temporal information from clips. Sadly, they conclude that temporal pooling of convoluted features proved more effective than LSTM stacked after trained feature maps. In the current paper, authors build on the same idea of using LSTM blocks (decoder) after convolution blocks(encoder) but using end-to-end training of entire architecture. They also compared RGB and optical flow as input choice and found that a weighted scoring of predictions based on both inputs was the best.

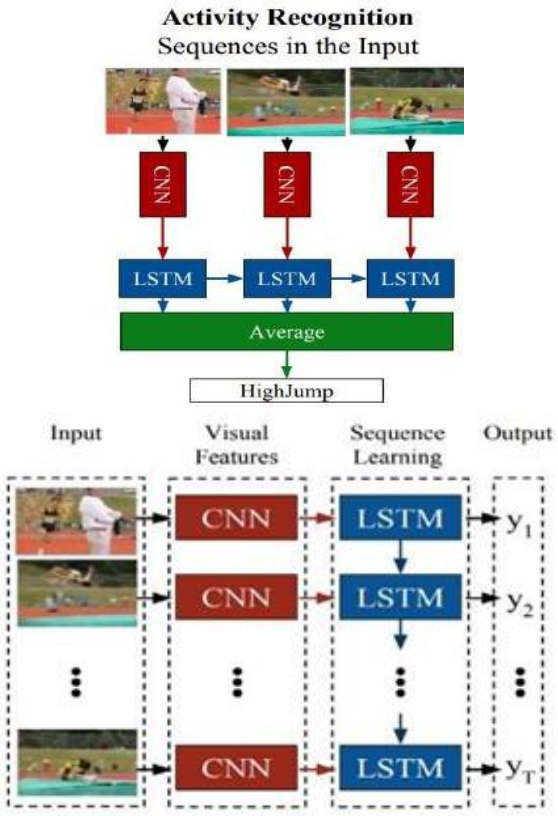

Fig 5: Left: LRCN for action recognition. Right: Generic LRCN architecture for all tasks.

Algorithm:

During training, 16 frame clips are sampled from video. The architecture is trained end-to- end with input as RGB or optical flow of 16 frame clips. Final prediction for each clip is the average of predictions across each time step. The final prediction at video level is average of predictions from each clip.

Benchmarks (UCF101-split1):

Score	Comment
82.92	Weighted score of flow and RGB inputs
71.1	Score with just RGB

My comments:

Even though the authors suggested end-to-end training frameworks, there were still a few drawbacks

- False label assignment as video was broken to clips
- Inability to capture long range temporal information
- Using optical flow meant pre-computing flow features separately

Varol et al. in their work[10] tried to compensate for the stunted temporal range problem by using lower spatial resolution of video and longer clips (60 frames) which led to significantly better performance.

C3D

- Learning Spatiotemporal Features with 3D Convolutional Networks
- Du Tran et al.
- Submitted on 02 December 2014
- Arxiv Link

Key Contributions:

- Repurposing 3D convolutional networks as feature extractors
- Extensive search for best 3D convolutional kernel and architecture
- Using deconvolutional layers to interpret model decision

Explanation:

In this work authors built upon work by Karpathy et al. However, instead of using 2D convolutions across frames, they used 3D convolutions on video volume. The idea was to train these vast networks on Sports1M and then use them (or an ensemble of nets with different temporal depths) as feature extractors for other datasets. Their finding was a simple linear classifier like SVM on top of ensemble of extracted features worked better than she ttate- of-the-art algorithms. The model performed even better if hand crafted features like iDT were used additionally.

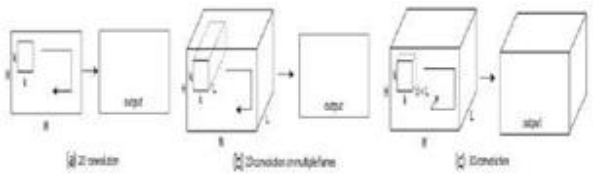

Fig 6: Differences in C3D paper and single stream paper.

The other interesting part of the work was using deconvolutional layers (explained here) to interpret the decisions. Their finding was that the net focussed on spatial appearance in first few frames and tracked the motion in the subsequent frames.

Algorithm:

During training, five random 2-second clips are extracted for each video with ground truth as action reported in the entire video. In test time, 10 clips are randomly sampled and predictions across them are averaged for final prediction.

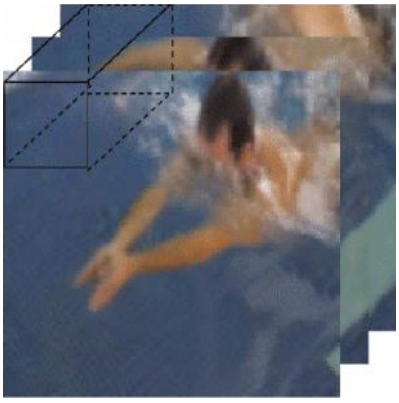

Fig 7: 3D convolution where convolution is applied on a spatiotemporal cube.

Score	Comment
82.3	C3D (1 net) + linear SVM
85.2	C3D (3 nets) + linear SVM
90.4	C3D (3 nets) + iDT + linear SVM

My comments:

The long range temporal modeling was still a problem. Moreover, training such huge networks is computationally a problem - especially for medical imaging where pre- training from natural images doesn't help a lot.

Note: Around the same time Sun et al.[11] introduced the concept of factorized 3D conv networks (FSTCN), where the authors explored the idea of breaking 3D convolutions into spatial 2D convolutions followed by temporal 1D convolutions. The 1D convolution, placed after 2D conv layer, was implemented as 2D convolution over temporal and channel dimension. The factorized 3D convolutions (FSTCN) had comparable results on UCF101 split.

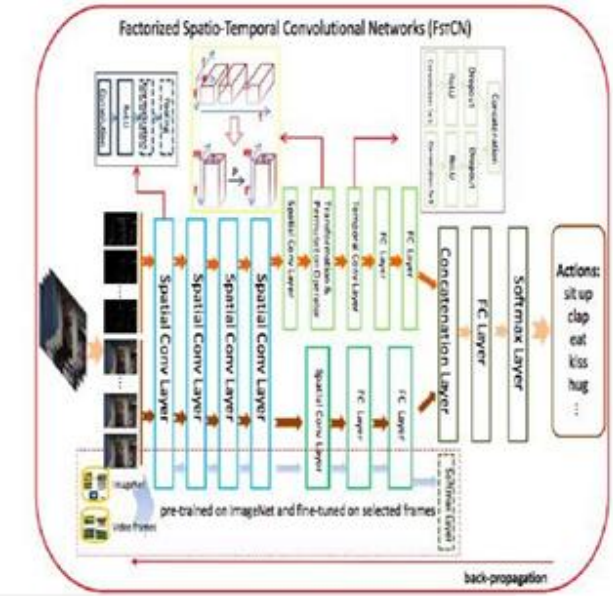

Fig 8. Schmatic diagram of Estcn foraction recognition

FSTCN paper and the factorization of 3D convolution Source.

Conv3D & Attention

- Describing Videos by Exploiting Temporal Structure
- Yao et al.
- Submitted on 25 April 2015
- Arxiv Link

Key Contributions:

- Novel 3D CNN-RNN encoder-decoder architecture which captures local spatiotemporal information
- Use of an attention mechanism within a CNN-RNN encoder-decoder framework to capture global context

Explanation:

Although this work is not directly related to action recognition, but it was a landmark work in terms of video representations. In this paper the authors use a 3D CNN + LSTM as base architecture for video description task. On top of the base, authors use a pre-trained 3D CNN for improved results.

Algorithm:

The set up is almost same as encoder-decoder architecture described in LRCN with two differences

1. Instead of passing features from 3D CNN as is to LSTM, 3D CNN feature maps for the clip are concatenated with stacked 2D feature maps for the same set of frames to enrich representation {v1, v2, …, vn} for each frame
   i. Note: The 2D & 3D CNN used is a pre- trained one and not trained end-to-end like LRCN
2. Instead of averaging temporal vectors across all frames, a weighted average is used to combine the temporal features. The attention weights are decided based on LSTM output at every time step.

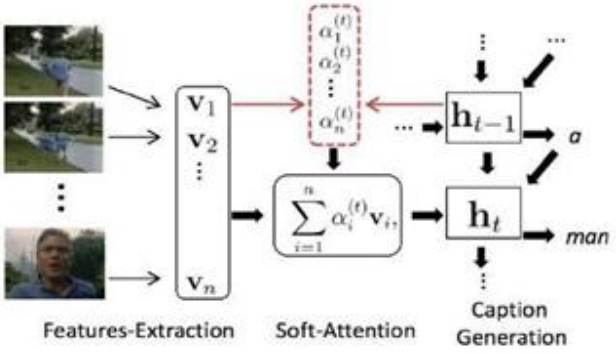

Fig 9: Attention mechanism for action recognition.

Benchmarks:

Score	Comment
–	Network used for video description prediction

My comments:

This was one of the landmark work in 2015 introducing attention mechanism for the first time for video representations.

TwoStreamFusion

- Convolutional Two-Stream Network Fusion for Video Action Recognition
- Feichtenhofer et al.
- Submitted on 22 April 2016
- Arxiv Link

Key Contributions:

- Long range temporal modeling through better long range losses
- Novel multi-level fused architecture

Explanation:

In this work, authors use the base two stream architecture with two novel approaches and demonstrate performance increment without any significant increase in size of parameters. The authors explore the efficacy of two major ideas.

1. Fusion of spatial and temporal streams (how and when) - For a task discriminating between brushing hair and brushing teeth - spatial net can capture the spatial dependency in a video (if it's hair or teeth) while temporal net can capture presence of periodic motion for each spatial location in video. Hence it's important to map spatial feature maps pertaining to say a particular facial region to temporal feature map for the corresponding region. To achieve the same, the nets need to be fused at an early level such that responses at the same pixel position are put in correspondence rather than fusing at end (like in base two stream architecture).
2. Combining temporal net output across time frames so that long term dependency is also modeled.

Algorithm:

Everything from two stream architecture remains almost similar except

1. As described in the figure below, outputs of conv_5 layer from both streams are fused by conv+pooling. There is yet another fusion at the end layer. The final fused output was used for spatiotemporal loss evaluation.

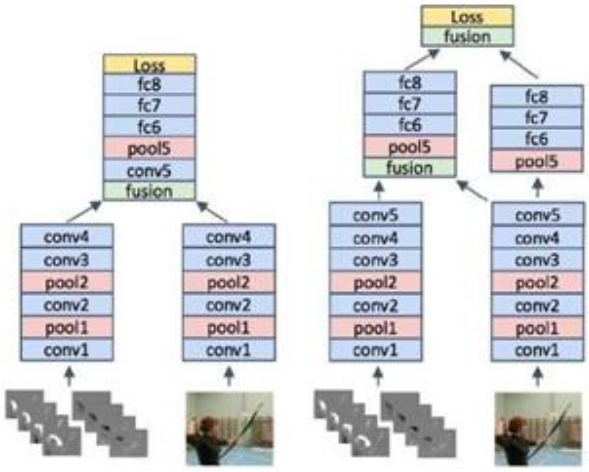

Fig 10: Possible strategies for fusing spatial and temporal streams. The one on right performed better.

2. For temporal fusion, output from temporal net, stacked across time, fused by conv+pooling was used for temporal loss

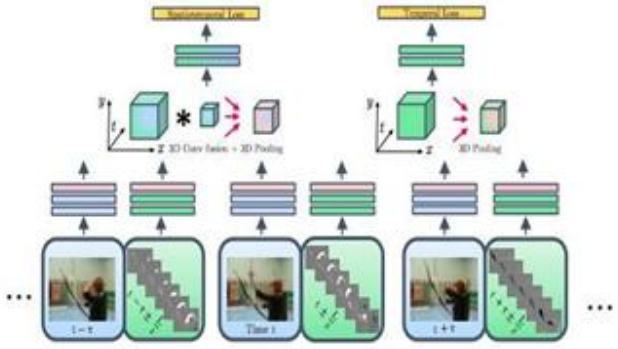

Fig 11: Two stream fusion architecture. There are two paths one for step 1 and other for step 2.

Benchmarks (UCF101-split1):

Score	Comment
92.5	TwoStreamfusion
94.2	TwoStreamfusion + iDT

My comments: The authors established the supremacy of the TwoStreamFusion method as it improved the performance over C3D without the extra parameters used in C3D.

TSN

- Temporal Segment Networks: Towards Good Practices for Deep Action Recognition
- Wang et al.
- Submitted on 02 August 2016
- Arxiv Link

Key Contributions:

- Effective solution aimed at long range temporal modeling
- Establishing the usage of batch normalization, dropout and pre-training as good practices

Explanation:

In this work authors improved on two streams architecture to produce state-of-the-art results. There were two major differences from the original paper

1. They suggest sampling clips sparsely across the video to better model long range temporal signal instead of the random sampling across entire video.
2. For final prediction at video-level authors explored multiple strategies. The best strategy was:
   a. Combining scores of temporal and spatial streams (and other streams if other input modalities are involved) separately by averaging across snippets
   b. Fusing score of final spatial and temporal scores using weighted average and applying softmax over all classes.

The other important part of the work was establishing the problem of overfitting (due to small dataset sizes) and demonstrating usage of now-prevalent techniques like batch normalization, dropout and pre-trainign to counter the same. The authors also evaluated two new input modalities as alternate to optical flow - namely warped optical flow and RGB difference.

Algorithm:

During training and prediction a video is divided into K segments of equal durations. Thereafter, snippets are sampled randomly from each of the K segments. Rest of the steps remained similar to two stream architecture with changes as mentioned above.

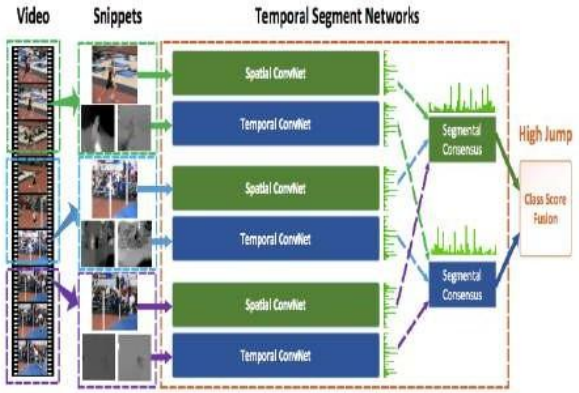

Fig 12: Temporal Segment Network architecture.

Benchmarks (UCF101-split1):

Score	Comment
94.0	TSN (input RGB + Flow)
94.2	TSN (input RGB + Flow + Warped flow)

My comments:

The work attempted to tackle two big challenges in action recognition - overfitting due to small sizes and long range modeling and the results were really strong. However,the problem of pre-computing optical flow and related input modalities was still a problem at large.

ActionVLAD

- ActionVLAD: Learning spatio-temporal aggregation for action classification
- Girdhar et al.
- Submitted on 10 April 2017
- Arxiv Link

Key Contributions:

- Learnable video-level aggregation of features
- End-to-end trainable model with video- level aggregated features to capture long term dependency

Explanation:

In this work, the most notable contribution by the authors is the usage of learnable feature aggregation (VLAD) as compared to normal aggregation using maxpool or avgpool. The aggregation technique is akin to bag of visual words. There are multiple learned anchor-point (say c1, …ck) based vocabulary representing k typical action (or sub-action) related spatiotemporal features. The output from each stream in two stream architecture is encoded in terms of k-space “action words” features - each feature being difference of the output from the corresponding anchor-point for any given spatial or temporal location.

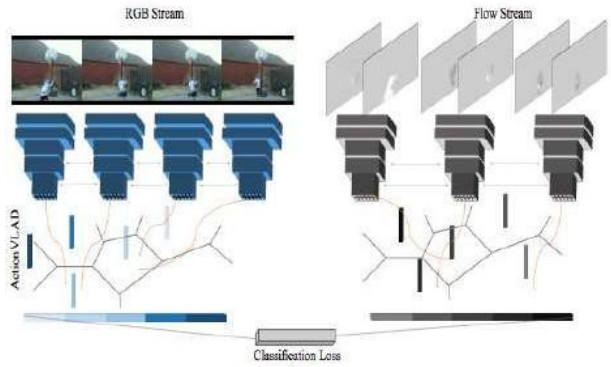

Fig 13: ActionVLAD - Bag of action based visual "words".

Average or max-pooling represent the entire distribution of points as only a single descriptor which can be sub-optimal for representing an entire video composed of multiple sub-actions. In contrast, the proposed video aggregation represents an entire distribution of descriptors with multiple sub-actions by splitting the descriptor space into k cells and pooling inside each of the cells.

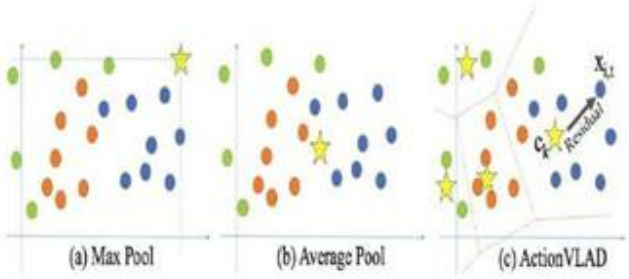

Fig 14

While max or average pooling are good for similar features, they do not not adequately capture the complete distribution of features. ActionVlAD clusters the appearance and motion features and aggregates their residuals from nearest cluster centers. Source.

Algorithm:

Everything from two stream architecture remains almost similar except the usage of ActionVLAD layer. The authors experiment multiple layers to place ActionVLAD layer
with the late fusion after conv layers working out as the best strategy.

Benchmarks (UCF101-split1):

Score	Comment
92.7	ActionVLAD
93.6	ActionVLAD + iDT

My comments:

The use of VLAD as an effective way of pooling was already proved long back. The extension of the same in an end-to-end trainable framework made this technique extremely robust and state-of-the-art for most action recognition tasks in early 2017.

HiddenTwoStream

- Hidden Two-Stream Convolutional Networks for Action Recognition
- Zhu et al.
- Submitted on 2 April 2017
- Arxiv Link

Key Contributions:

- Novel architecture for generating optical flow input on-the-fly using a separate network

Explanation:

The usage of optical flow in the two stream architecture made it mandatory to pre-compute optical flow for each sampled frame before hand thereby affecting storage and speed adversely. This paper advocates the usage of an unsupervised architecture to generate optical flow for a stack of frames.

Optical flow can be regarded as an image reconstruction problem. Given a pair of adjacent frames I1 and I2 as input, our CNN generates a flow field V. Then using the predicted flow field V and I2, I1 can be reconstructed as I1' using inverse warping such that difference between I1 and it's reconstruction is minimized.

Algorithm:

The authors explored multiple strategies and architectures to generate optical flow with largest fps and least parameters without hurting accuracy much. The final architecture was same as two stream architecture with changes as mentioned:

1. The temporal stream now had the optical flow generation net (MotionNet) stacked on the top of the general temporal stream architectures. The input to the temporal stream was now consequent frames instead of preprocessed optical flow.
2. There's an additional multi-level loss for the unsupervised training of MotionNet

The authors also demonstrate improvement in performance using TSN based fusion instead of conventional architecture for two stream approach.

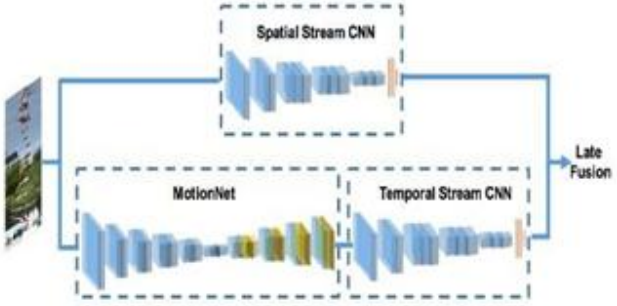

Fig 15: HiddenTwoStream - MotionNet generates optical flow on-the-fly.

Benchmarks (UCF101-split1):

Score	Comment
89.8	Hidden Two Stream
92.5	Hidden Two Stream + TSN

My comments:

The major contribution of the paper was to improve speed and associated cost of prediction. With automated generation of flow, the authors relieved the dependency on slower traditional methods to generate optical flow.

I3D

- Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset
- Carreira et al.
- Submitted on 22 May 2017
- Arxiv Link

Key Contributions:

- Combining 3D based models into two stream architecture leveraging pre-training
- Kinetics dataset for future benchmarking and improved diversity of action datasets

Explanation:

This paper takes off from where C3D left. Instead of a single 3D network, authors use two different 3D networks for both the streams in the two stream architecture. Also, to take advantage of pre-trained 2D models the authors repeat the 2D pre-trained weights in the 3rd dimension. The spatial stream input now consists of frames stacked in time dimension instead of single frames as in basic two stream architectures.

Algorithm:

Same as basic two stream architecture but with 3D nets for each stream.

Benchmarks (UCF101-split1):

Score	Comment
93.4	Two Stream I3D
98.0	Imagenet + Kinetics pre-training

My comments:

The major contribution of the paper was the demonstration of evidence towards benefit of using pre-trained 2D conv nets. The Kinetics dataset, that was open-sourced along the paper, was the other crucial contribution from this paper.

T3D

- Temporal 3D ConvNets: New Architecture and Transfer Learning for Video Classification
- Diba et al.
- Submitted on 22 Nov 2017
- Arxiv Link

Key Contributions:

- Architecture to combine temporal information across variable depth
- Novel training architecture & technique to supervise transfer learning between 2D pre- trained net to 3D net

Explanation:

The authors extend the work done on I3D but suggest using a single stream 3D DenseNet based architecture with multi-depth temporal pooling layer (Temporal Transition Layer) stacked after dense blocks to capture different temporal depths The multi depth pooling is achieved by pooling with kernels of varying temporal sizes.

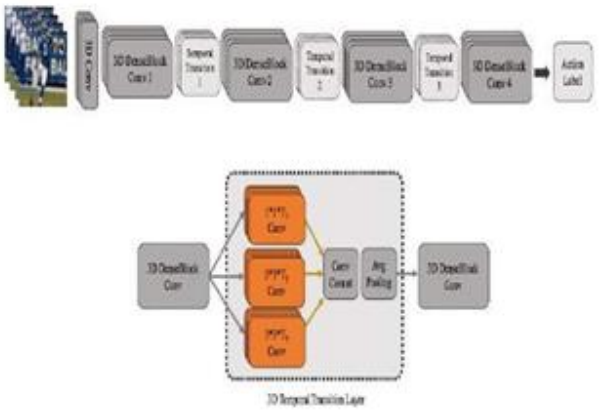

Fig 16: TTL Layer along with rest of DenseNet architecture.

Apart from the above, the authors also devise a new technique of supervising transfer learning betwenn pre-trained 2D conv nets and T3D. The 2D pre-trianed net and T3D are both presented frames and clips from videos where the clips and videos could be from same video or not. The architecture is trianed to predict 0/1 based on the same and the error from the prediction is back-propagated through the T3D net so as to effectively transfer knowledge.

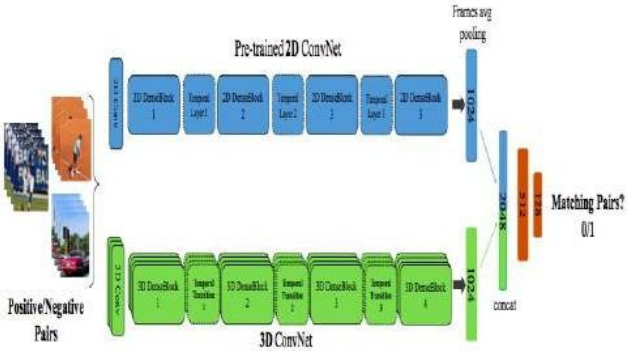

Fig 17: Transfer learning supervision. Source.

Algorithm:

The architecture is basically 3D modification to DenseNet [12] with added variable temporal pooling.

Benchmarks (UCF101-split1):

Score	Comment
90.3	T3D
91.7	T3D + Transfer
93.2	T3D + TSN

My comments:

Although the results don't improve on I3D results but that can mostly attributed to much lower model footprint as compared to I3D. The most novel contribution of the paper was the supervised transfer learning technique.

REFERENCES

[1] ConvNet Architecture Search for Spatiotemporal Feature Learning by Du Tran et al.
[2] Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset
[3] Action recognition by dense trajectories by Wang et. al.
[4] On space-time interest points by Laptev
[5] Behavior recognition via sparse spatio- temporal features by Dollar et al
[6] Action Recognition with Improved Trajectories by Wang et al.
[7] 3D Convolutional Neural Networks for Human Action Recognition by Ji et al.
[8] Large-scale Video Classification with Convolutional Neural Networks by Karpathy et al.
[9] Beyond Short Snippets: Deep Networks for Video Classification by Ng et al.
[10] Long-term Temporal Convolutions for Action Recognition by Varol et al.
[11] Human Action Recognition using Factorized Spatio-Temporal Convolutional Networks by Sun et al.
[12] Densely Connected Convolutional Networks by Huang et al.