Research & Development
–DRAFT–

# Evaluation of current Approaches for Situation-Awareness in Autonomous Systems from Action Recognition in Video Data

*Author:*
Maximilian Schöbel

*Advisors:*
Prof. Dr. Erwin Prassler
Prof. Dr. Paul G. Plöger

January 12th, 2017

# Contents

# 1 Introduction

META: Brief but concise review of the first two "W"s.

The operation of autonomous mobile systems in public, uncontrolled environments is despite active research still a difficult task.

Humans are perfectly able to act and move in unknown, crowded environments and even react successfully to new situations because they are aware of their surroundings.

An important part of Situation Awareness is the knowledge of what actions are currently performed by persons in the vicinity of an agent. This knowledge enables the agent to derive a suitable policy for its own future actions.

Actions of interest are single-person actions, person-person interactions, person-object interactions and group activities.

Enabling situation-awareness in autonomous systems is an important goal, which has an impact on other problems in autonomous systems.

Possible applications:
Pedestrian movement predicition in robotics,
Risk and danger evaluation through video surveillance in public environements,
surveillance of children or the elderly in assisted living environments,
patient monitoring in hospitals,
video retrieval (content-based video indexing),
human-computer interaction.

Requirement: Automated recognition of high-level actions.

Situation awareness is an abstract concept, which includes lots of independent manifestations and involves multiple sensory inputs.

This work focuses on approaches that process time-sequential video data, because video-cameras represent a cost-effective and widely used technology in many existing systems.

Motivation for using videos: Promising results in classfication tasks from images. Video adds another (temporal) dimension, which conveys a lot of information that can be accessed for classification as well. Video provides natural data augmentation cite:simnoyan two-stream paper (??)

## 1.1 Situation Awareness from video data

META: General Definition of Situation Awareness in the context of autonomous systems. Placement of Action Recognition among other vision-based methods, i.e. Action Prediction, Anomaly Detection, Event and Action Detection, Person/Pedestrian Detection, Gesture Recognition. Definitions of the above methods.

Simple case: Video contains the performance of a single human action which needs to be classified into one of several preknown classses.

General real-world case: System operates on a video stream and needs to perform continuous recognition of human actions, including detection of beginning and endings times of containing acions.

General Processing Pipeline for Action Recognition: Person Detection -> Tracking -> Action Detection -> Segmentation -> Action recognition.

Action Recognition: A part of Computer Vision research, it's goal is to automatically analyze human actions/actitvities from video-data.

Other sensory input than video possible

## 1.2 The Action Recognition Problem

Action Recognition is a classification-task.

Difference to face/gate recognition: Generalize over person characteristics.

Intra- and inter-class variances.

Background and recording settings.

Temporal variations.

Obtaining and labeling training data.

Main task of action recognition research: Overcome these challenges and built systems, that recognize actions robustly, even when performed by different persons in differently lighted environments at different speeds.

Main components: (i) A discriminative architecture that is able to recognise the general characteristics of different action classes while ignoring personal characteristics of different performers. (ii) Large datasets that provide this information by containing many different examples for each action class.

## 1.3 Survey Papers in Action Recognition (Related work)

Review of most important/recent review papers in Action Recognition with traditional and Deep Learning approaches.

### 1.3.1 A survey on vision-based human action recognition, Ronald Poppe (2010)

**Definition of action:** Uses the hierarchical classification of human motion in action primitives, actions and activities as given in Moeslund et al. (cite ??)

Action primitives are atomic movements at the limb-level.

Actions are possibly cyclic whole body movements and consist of multiple action primitives.

Activities consist of multiple actions whose subsequent execution make the movement interpretable.

Example: Action primitives: Left/right leg forward -> Action: Starting, Running, Jumping -> Activity: Jumping hurdles.

**Scope:** Gives a very good classification of conventional methods in human action recognition.

The discussion is split according to video representations and classification methods.

Challenges of the domain are described very well.

**Deficits:** No Deep Learning methods are discussed.

Datasets and benchmarks are only discussed briefly.

### 1.3.2 Human Activity Analysis: A Review – Aggarwal and Ryoo (2011)

Gives an approach-based taxonomy.

### 1.3.3 A survey on vision-based methods for action representation, segmentation and recognition – Weinland et al. (2011)

### 1.3.4 A survey of video datasets for human action and activity recognition – Chaquet et al. (2013)

### 1.3.5 A review of unsupervised feature learning and deep learning for time-series modeling – Längkvist et al. (2014)

### 1.3.6 Going Deeper into Action Recognition: A survey – Herath et al. (2016)

**Definition of action:**

# 2 Conventional Methods in Action Recognition

META: Condensed overview and description of conventional Methods in action Recognition using the taxonomy of Aggarwal and Ryoo's fine survey paper. More detailed description of methods using local-features, since these have become the standard approach in action recognition after Aggarwal and Ryoo's overview.

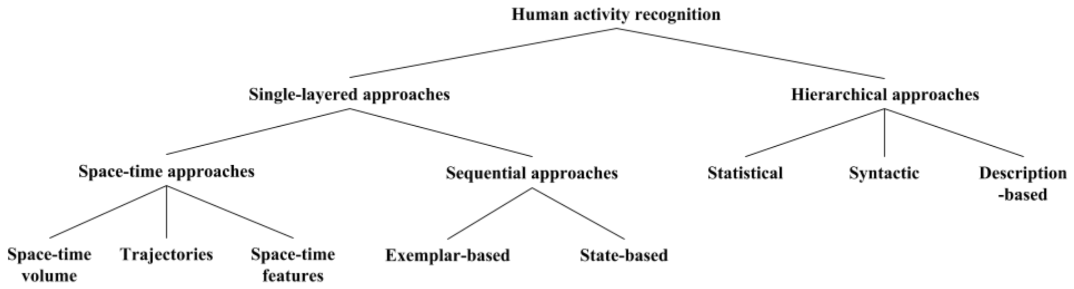


*Figure 1:* Approach-based taxonomy for conventional methods in human activity recognition as given by Aggarwal and Ryoo[1]

3 Main components in action recognition using local features: Feature Extraction, Representation Building, Classification.

Methods for feature extraction: Interest point detectors or dense sampling.

Space-time interest point detectors: Harris3D[2], Cuboids[3], Hessian Detector[4]

Descriptors for 3D volumes around previously detected space-time interest points: Histogram of Gradient HOG[5], Histogram of Optical Flow (HOF)[6], 3D Histogram of Gradient (HOG3D)[7], Extended SURF (ESURF)[4]

# 3 Deep Learning Methods in Action Recognition

META: Review of approaches that use Deep Learning.

## 3.1 Spatio-Temporal Networks

I.e. convolutional methods.

### 3.1.1 3D Convolutional Neural Networks for Human Action Recognition – Ji et al. (2013)

In this work [8] the authors porpose 3D convolution for action recognition from video data, which processes spatial as well as temporal information in a convolutional layer.

In regular convolutional neural networks 2D convolutions are applied in the convolutional layers to extract features from the feature maps in the previous layer. More formally in the notation of the authors, the value of feature map $j$ in the $i$th layer at spatial position $(x, y)$ is given by:

$$v_{ij}^{xy} = \tanh\left(b_{ij} + \sum_{m} \sum_{p=0}^{P_i-1} \sum_{q=0}^{Q_i-1} w_{ijm}^{pq} v_{(i-1)m}^{(x+p)(y+q)}\right)$$

$w_{ijm}^{pq}$ denotes the value at position $(p, q)$ of the kernel, that performs convolution on feature map $m$ of the previous layer, resulting in feature map $j$ in layer $i$. $P_i$ and $Q_i$ denote the dimensions of the kernel in x- and y-direction respectively. Tensor $w$ therefore represents all kernels that produce the feature maps in layer $i$ through convolution.

The two inner sums carry out the convolutional operation on feature map $m$ of the previous layer, which is then combined with the results for the other feature maps by summation over $m$, added with a bias and fed into a non-linear function $(\tanh(\cdot))$ to result in the value of the current feature map.

Note: The convolutional operation used here is called cross-correlation, which differs from mathematical discrete convolutions in that the kernel is not flipped. This results in a non-commutative operation as described in chapter 9 of cite deep learning book ??

The authors propose an extension of 2D convolutions by using a three dimensional kernel. More formally, in the notation as above, the value of feature map $j$ in the $i$th layer at position $(x, y, z)$ is given by:

$$v_{ij}^{xyz} = \tanh\left(b_{ij} + \sum_{m}\sum_{p=0}^{P_i-1}\sum_{q=0}^{Q_i-1}\sum_{r=0}^{R_i-1} w_{ijm}^{pqr} v_{(i-1)m}^{(x+p)(y+q)(z+r)}\right)$$

As above, $w_{ijm}^{pqr}$ denotes the value of the now three dimensional kernel at position $(p, q, r)$, which performs convolution on the $m$th feature map of the previous layer. $R_i$ denotes the dimension of the kernel in temporal direction.

Based on the 3D convolution, the authors design a neural network architecture, that takes an input of 7 frames of size 60x40.

The network is evaluated as part of an action detection and recognition system on the TRECVID (TREC Video Retrieval Evaluation) data, which consists of surveillance videos recorded at London Gatwick Airport.

The details of the architecture are described below:

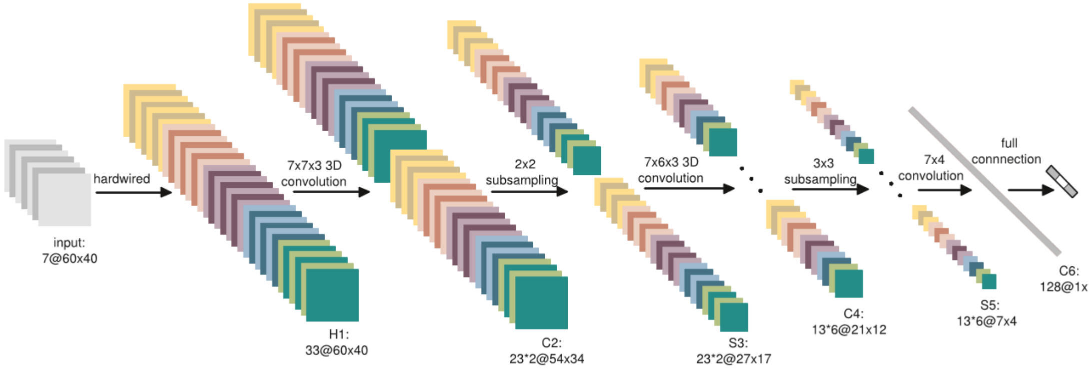


*Figure 2:* 3D CNN architecture developed for human action recognition on the TRECVID 2008 development dataset [8]

At first several hard wired kernels are applied to the input frames, which extract gray values, gradients along the horizontal and vertical direction in the frames and the optical flow between two consecutive frames. This results in 33 feature maps, organized in 5 different channels: gray, gradient-x, gradient-y, optflow-x and optflow-y.

In the first convolutional layer C2 two 3D kernels with dimesions 7x7x3 (7x7 in the spatial dimension and 3 in the temporal dimension) are applied at each of the five channels separately. The first convolutional layer therefore contains $2 \times 5 \times 7 \times 7 \times 3$ (kernel-weights) $+$ $2 \times 5$ (biases) $= 1480$ trainable parameters.

After a subsampling layer S3, three different kernels are applied to each of the 2x5 channels of the previous layer.

The last convolutional layer performs 2D convolutions to obtain 128 feature maps of dimension 1x1. These feature maps are a 128 dimensional vector representation of the input.

The resulting feature vector is then classified (in this task here into three different classes) by a fully connected layer.

The complete architecture has 295.458 trainable parameters in total, which are initialized randomly and learned by online error back-propagation. cite lecun 1998

### 3.1.2 Large-scale Video Classification with Convolutional Neural Networks – Karpathy et al. (2014)

## 3.2 Multiple Stream Networks

The most successfull architecture at action recognition. They are equally powerful as the improved dense trajectories approach. cite TDD ??

These approaches use the decomposability of videos into a spatial component (analysing different frames) and a temporal component (analysing the change between frames) for action recognition.

The first architecture of this kind was proposed in 2014 from Simonyan and Zisserman.

### 3.2.1 Two-Stream Convolutional Networks for Action Recognition in Videos - Simonyan and Zisserman (2014)

The authors propose a novel architecture for action recognition with two separate recognition streams (spatial- and temporal-stream) which are combined by late fusion.

The authors evaluate two different fusion methods: building the average of both network's outputs and training a linear multi-class SVM on stacked $L_2$-normalised softmax scores.

This approach is motivated by the two-streams hypothesis cite (??), according to which the human visual cortex contains two paths: the ventral stream for object recognition and the dorsal stream for recognising motion.

Both streams are implemented as deep CNNs, with the rectification activation function for all hidden units.

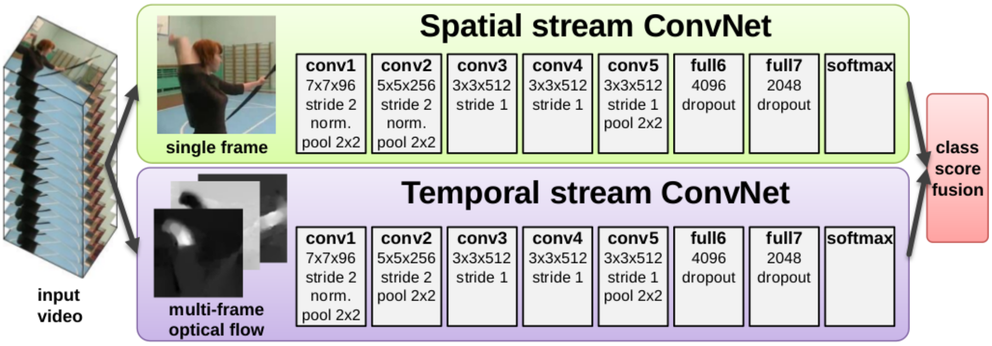


*Figure 3:* Two-stream architecture for video classification, depicting the spatial- and temporal-stream with implementation details as deep Convolutional Neural Network [9]

Spatial stream: Takes single video frames as input. Performs action recognition from still images and is fairly competitive on its own. Basically an image recognition architecture. Advantage: Can be pre-trained using large amount of image data, here from the ImageNet challenge dataset cite (??).

Spatial part of a video, i.e. the individual static frames, convey information about the objects and persons in the scene.

Temporal stream: is trained to recognize actions from motions given in the form of dense optical flow.

The temporal part of a video, i.e. the change between frames, conveys information about the movement of the observer (camera) and the movement of objects in the scene.

The second normalisation layer was removed from the the temporal stream network in order to reduce memory usage.

The authors propose two methods for constructing the input to the temporal network by stacking optical flow displacement fields along several consecutive frames of the input video.

**Optical Flow Stacking:**
A dense optical flow field $\mathbf{d}_t(u, v)$ of two consecutive frames at times $t$ and $t + 1$ can be thought of as a two dimensional vector-field, which maps the displacement of each pixel along the transition from frame $t$ to $t+1$. In this case $u, v \in \mathbb{N}$, $1 \leq u \leq w$ and $1 \leq v \leq h$ where $w$ and $h$ are the width and height of the video frames.

The horizontal and vertical components $d_t^x(u, v)$, $d_t^y(u, v)$ can be interpreted as image channels.

This method constructs the input volume $I_\tau \in \mathbb{R}^{w \times h \times 2L}$ of the temporal stream network by stacking the horizontal and vertical components of the dense optical flow field along

$L$ consecutive frames, beginning at time $\tau$. Formally, with $1 \leq k \leq L$:

$$I_\tau(u, v, 2k-1) = d^x_{\tau+k-1}(u, v)$$
$$I_\tau(u, v, 2k) = d^y_{\tau+k-1}(u, v)$$

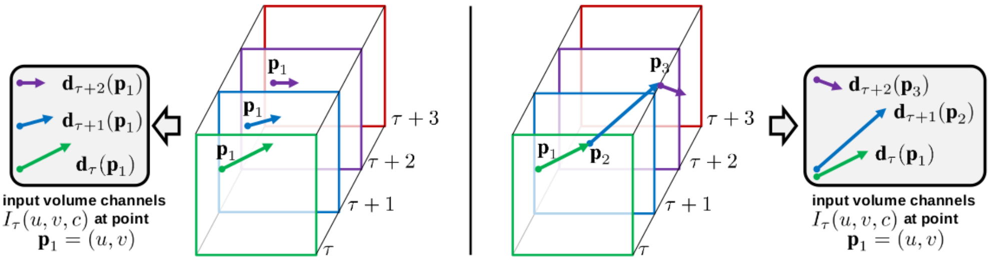
**Trajectory Stacking:**
Instead of sampling at fixed locations in each frame, this methods samples the dense optical flow field along the motion trajectories of the initial points in frame $\tau$.

Let $\mathbf{p}_k$ denote the motion trajectory of initial point $(u, v)$. With $1 < k \leq L$ and $\mathbf{p}_1 = (u, v)$ the trajectory is recursively defined by:

$$\mathbf{p}_k = \mathbf{p}_{k-1} + \mathbf{d}_{\tau+k-2}(\mathbf{p}_{k-1})$$

The input volume can then be constructed by sampling the horizontal and vertical optical flow components along these trajectories.

$$I_\tau(u, v, 2k-1) = d^x_{\tau+k-1}(\mathbf{p}_k)$$
$$I_\tau(u, v, 2k) = d^y_{\tau+k-1}(\mathbf{p}_k)$$



$\mathbf{d}_{\tau+2}(\mathbf{p}_1)$
$\mathbf{d}_{\tau+1}(\mathbf{p}_1)$
$\mathbf{d}_{\tau}(\mathbf{p}_1)$
input volume channels
$I_\tau(u, v, c)$ at point
$\mathbf{p}_1 = (u, v)$
$\mathbf{p}_1$
$\tau+3$
$\tau+2$
$\tau+1$
$\tau$
$\mathbf{p}_3$
$\mathbf{p}_2$
$\mathbf{d}_{\tau+2}(\mathbf{p}_3)$
$\mathbf{d}_{\tau+1}(\mathbf{p}_2)$
$\mathbf{d}_{\tau}(\mathbf{p}_1)$

*Figure 4:* Construction of input volumes from multi-frame optical flow. Left: Optical Flow Stacking. Right: Trajectory Stacking [9]

Since the Convolutional Networks require fixed sized inputs, the authors sample a $224 \times 224 \times 2L$ subvolume from $I_\tau \in \mathbb{R}^{w \times h \times 2L}$ and use it as the temporal networks input.

By using optical flow, the authors explicitly incorporate a representation of motion in their action recognition architecture.

The authors further describe two optional techniques that are evaluated for constructing the inputs with either one of optical flow stacking methods.

1. **Bi-directional Optical Flow**: The input volume $I_\tau$ is created by using regular forward optical flow from frame $\tau$ to $\tau + L/2$ and additionally calculated optical flow from frame $\tau$ to $\tau - L/2$ in the backwards direction. Stacking the horizontal and vertical components of these optical flow fields results in an input volume of length $L/2$ around frame $\tau$ in both directions.

2. **Mean flow subtraction**: The displacement vectors between two frames can dominantly be caused by global movement of the camera. Compared to regular camera-motion compensation, the authors use a simpler approach and just subtract the mean vector from each displacement field $\mathbf{d}_t$.

An advantage of separating the spatial and the temporal stream is the possibility of pre-training the spatial net with large pre-existing image datasets (here ImageNet ILSVRC-2012 ??).

For training the temporal stream network with video-data, the authors address the general problem of video action recognition, that the existing datasets are too small in comparison to their image-dataset counterparts.

The authors use, according to the multi-task learning paradigm [10], the UCF-101 and the HMDB-51 dataset simultaneously for training the temporal stream network (see chapter ??).

By training a network on several tasks (here UCF-101 classification and HMDB-51 classification), the network learns more general video representations, since the second task acts as regulariser and more data can be utilized.

The authors implement this by using two softmax classification layers, one for each dataset. Each softmax layer has its own loss function and the sum of the individual losses is taken as the overall training loss for computing the weight updates by backpropagation.

Training for both networks is conducted with mini-batch gradient descent with 256 randomly selected videos at each iteration. From each of those videos, a single frame is randomly chosen, a $224 \times 224$ sub-image is randomly cropped, randomly horizontal flipped, RGB jittered and then used as training input for the spatial stream network.

An optical flow volume is constructed for this selected frame as described above. For optical flow computation the authors use a fast implementation (0.06s per pair of frames) from the OpenCV toolbox. Despite it's speed, on-the-fly computation of optical-flow would be a bottleneck and is therefore pre-computed and stored for the complete datasets.

The creators of UCF-101 and HMDB-51 provide three splits of their datasets into training- and testing-data. The standard evaluation procedure is to report the average accuracy over those three splits, which the authors follow in this work as well.

The authors build their final design of the two-stream architecture by evaluating different setups for the spatial and temporal stream network on their own using UCF-101 (split 1).

Besides using two different dropout rate (0.5 and 0.9), the performance of the spatial-stream network is evaluated for:

1. Training the complete network from scratch on UCF-101.
2. Pre-training the network on ILSVRC-2012 and fine-tuning it on UCF-101.

3. Pre-training the network on ILSVRC-2012 and fine-tuning of only the last (classification) layer.

For evaluating the temporal-stream network a fixed dropout rate of 0.9 is chosen, because the network needs to be trained from scratch. The performance is then measured for:

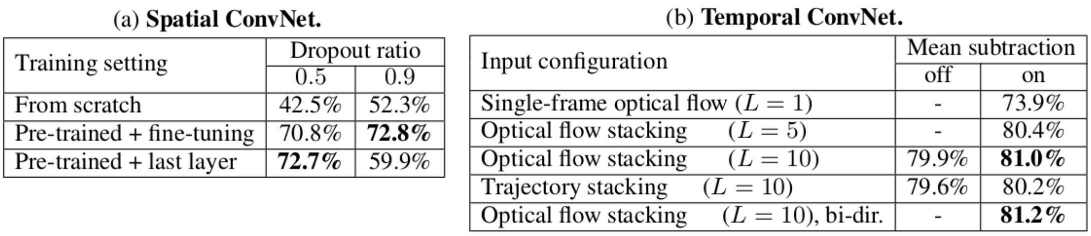
1. Using one or several (stacked) optical flow displacement fields $L \in 1, 5, 10$.
2. Regular optical flow stacking
3. Trajectory stacking
4. Using bi-directional optical flow
5. Using mean subtraction

The findings are presented in table 1.

Spatial-stream network: The authors decided on pre-trained the network on ILSVRC and fine-tuning only the last layer.

Temporal-stream network: Mean subtraction and stacking multiple optical flows is beneficial, so $L = 10$ is used as the default setting. Regular optical flow stacking performs better than trajectory stacking and bi-directional optical flow only yields slight improvement against forward optical flow. Therefore regular forwars optical flow stacking is chosen for the temporal-stream network.

The authors highlight that the temporal-stream network significantly outperforms the spatial-stream network, which confirms the importance of motion information for action recognition from video.



(a) **Spatial ConvNet.**

| Training setting | Dropout ratio | |
|---|---|---|
| | 0.5 | 0.9 |
| From scratch | 42.5% | 52.3% |
| Pre-trained + fine-tuning | 70.8% | **72.8%** |
| Pre-trained + last layer | **72.7%** | 59.9% |

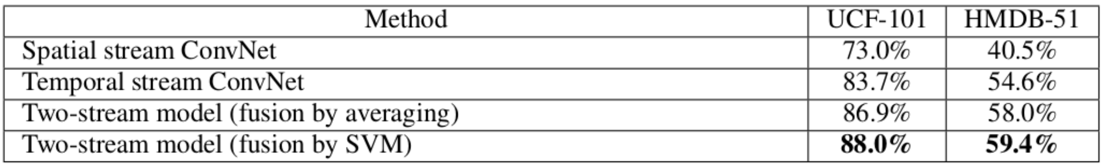
(b) **Temporal ConvNet.**

| Input configuration | Mean subtraction | |
|---|---|---|
| | off | on |
| Single-frame optical flow ($L = 1$) | - | 73.9% |
| Optical flow stacking ($L = 5$) | - | 80.4% |
| Optical flow stacking ($L = 10$) | 79.9% | **81.0%** |
| Trajectory stacking ($L = 10$) | 79.6% | 80.2% |
| Optical flow stacking ($L = 10$), bi-dir. | - | **81.2%** |

*Table 1:* Performance of the individual convolutional networks on UCF-101 (split 1) [9]

After having found the optimal configurations for the individual temporal-stream and spatial-stream networks, the authors evaluate different fusion methods (averaging and SVM) and find that fusion by SVM performs best. The fused network's performance significantly improves over the individual network's performance, which implies, that the spatial and temporal recognition stream are complementary.

The results in table 2 show, that fusion by SVM works best.



| Method | UCF-101 | HMDB-51 |
|---|---|---|
| Spatial stream ConvNet | 73.0% | 40.5% |
| Temporal stream ConvNet | 83.7% | 54.6% |
| Two-stream model (fusion by averaging) | 86.9% | 58.0% |
| Two-stream model (fusion by SVM) | **88.0%** | **59.4%** |

*Table 2:* Mean accuracy over three splits on UCF-101 and HMDB-51 [9]

### 3.2.2 Action recognition with trajectory-pooled deep-convolutional descriptors – Wang et al. (2014)

### 3.2.3 Fusing Multi-Stream Deep Networks for Video Classification – Wu et al. (2015)

## 3.3 Generative Models

Restricted Boltzmann Machine

## 3.4 Temporal Coherency Networks

# 4 Datasets and Benchmarks in Action Recognition

## 4.1 Review of Datasets for Human Action Classification

Review of the most important currently existing datasets, focus on newest ones (since 2013)

Reference dataset survey paper.

## 4.2 Data Augmentation

## 4.3 Alternative Benchmarks for Action Recognition Algorithms

## 4.4 Inter-Dataset Approaches

# 5 Evaluation

What do we need, what do we have, what is best suited so far?

## References

[1] Jake K. Aggarwal and Michael S. Ryoo. "Human Activity Analysis: A Review". In: *ACM Computing Surveys (CSUR)* 43.3 (2011). 01121, p. 16. URL: http://dl.acm.org/citation.cfm?id=1922653 (visited on 05/23/2016).

[2] Ivan Laptev. "On Space-Time Interest Points". In: *International Journal of Computer Vision* 64 (2-3 2005). 02614, pp. 107–123. URL: http://link.springer.com/article/10.1007/s11263-005-1838-7 (visited on 06/01/2016).

[3] Piotr Dollár et al. "Behavior Recognition via Sparse Spatio-Temporal Features". In: *Visual Surveillance and Performance Evaluation of Tracking and Surveillance, 2005. 2nd Joint IEEE International Workshop on.* 02076. IEEE, 2005, pp. 65–72. URL: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1570899 (visited on 05/16/2016).

[4] Geert Willems, Tinne Tuytelaars, and Luc Van Gool. "An Efficient Dense and Scale-Invariant Spatio-Temporal Interest Point Detector". In: *European Conference on Computer Vision.* 00647. Springer, 2008, pp. 650–663. URL: http://link.springer.com/chapter/10.1007/978-3-540-88688-4_48 (visited on 10/18/2016).

[5] Navneet Dalal and Bill Triggs. "Histograms of Oriented Gradients for Human Detection". In: *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05).* Vol. 1. 15268. IEEE, 2005, pp. 886–893. URL: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1467360 (visited on 07/18/2016).

[6] Ivan Laptev et al. "Learning Realistic Human Actions from Movies". In: *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on.* 02233. IEEE, 2008, pp. 1–8. URL: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4587756 (visited on 05/25/2016).

[7] Alexander Klaser, Marcin Marsza\lek, and Cordelia Schmid. "A Spatio-Temporal Descriptor Based on 3d-Gradients". In: *BMVC 2008-19th British Machine Vision Conference.* 00929. British Machine Vision Association, 2008, pp. 275–1. URL: https://hal.inria.fr/inria-00514853/ (visited on 10/18/2016).

[8] Shuiwang Ji et al. "3D Convolutional Neural Networks for Human Action Recognition". In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 35.1 (2013). 00485, pp. 221–231. URL: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6165309 (visited on 04/27/2016).

[9] Karen Simonyan and Andrew Zisserman. "Two-Stream Convolutional Networks for Action Recognition in Videos". In: *Advances in Neural Information Processing Systems.* 00353. 2014, pp. 568–576. URL: http://papers.nips.cc/paper/5353-two-stream-convolutional-networks-for-action-recognition-in-videos (visited on 05/06/2016).

[10] Ronan Collobert and Jason Weston. “A Unified Architecture for Natural Language Processing: Deep Neural Networks with Multitask Learning”. In: *Proceedings of the 25th International Conference on Machine Learning.* 01219. ACM, 2008, pp. 160–167. URL: `http://dl.acm.org/citation.cfm?id=1390177` (visited on 10/21/2016).