Computer Vision, Speech Communication & Signal Processing Group,
Intelligent Robotics and Automation Laboratory
Institute of Communication and Computer Systems (ICCS)
National Technical University of Athens, Greece (NTUA)


# Part 1:
# Spatio-Temporal Visual Processing

Petros Maragos and Petros Koutras

Tutorial at IEEE International Conference on Image Processing 2019,
Taipei, Taiwan, September 22, 2019

# Spatio-temporal Computer Vision problems

- **automatic video understanding** becomes one of the most essential and demanding challenges
- **static** computer vision problems:
  - image domain → no temporal evolution
    - object detection, semantic segmentation, pose estimation
  - deep learning and large datasets boosted the performance
- **spatio-temporal** problems:
  - video domain → related with the temporal information
    - spatio-temporal saliency, action recognition, video summarization
  - require the integration and modeling of the temporal evolution

# Action Recognition

- automated classification and detection of human activities on videos
  - action labels from human annotations
  - many large datasets: Hollywood2, UCF101, HMDB51, Kinetics

open car door

open door

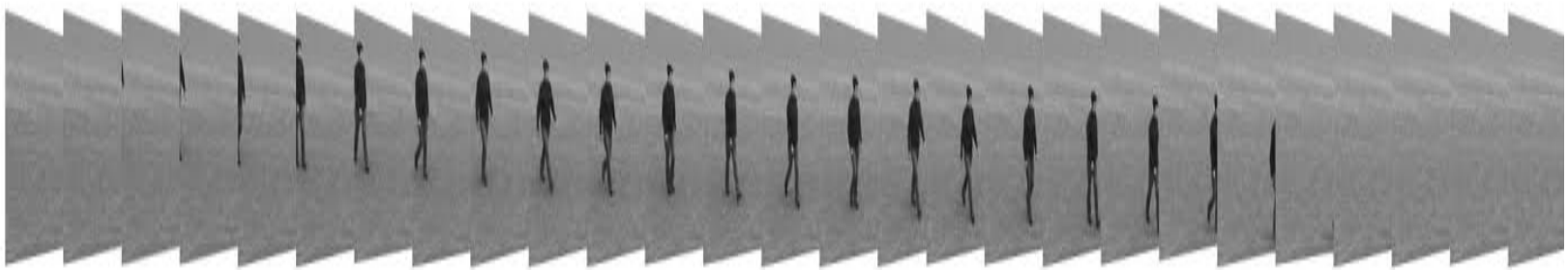
walk

talk

# Action Recognition Tasks

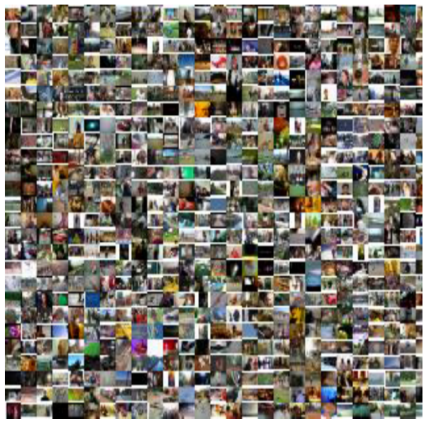
- classification
- detection

walking



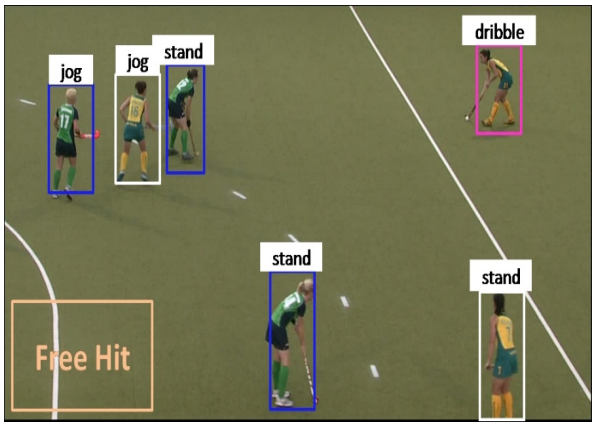
walking

# Video understanding applications



Indexing and analysis of big video data

Augmented reality and interactive video games

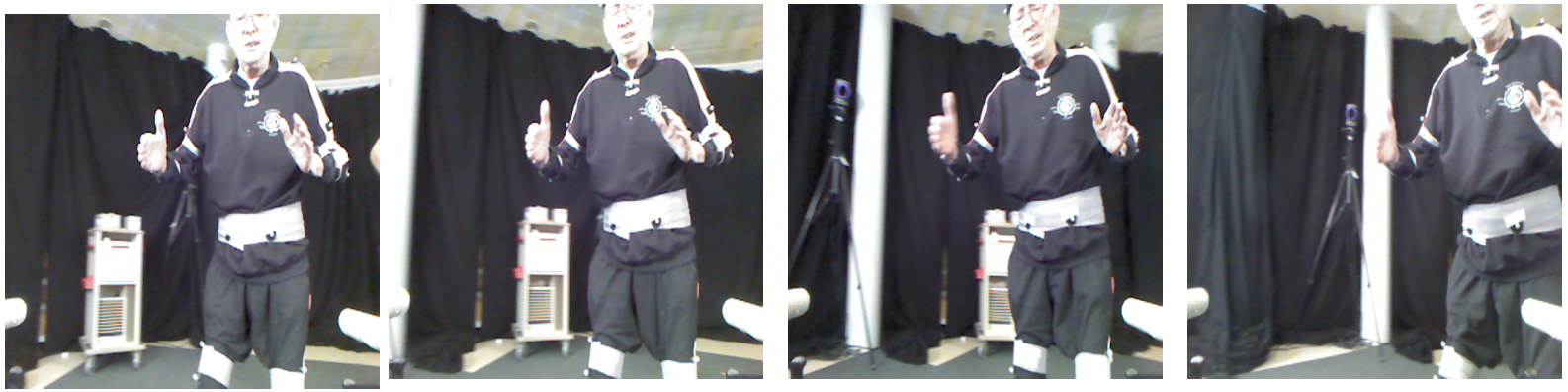



Sports analysis

# Video understanding robotic applications

Human-Robot Interaction (HRI)

Patient monitoring – assistive robotics



# Action Recognition - Challenges

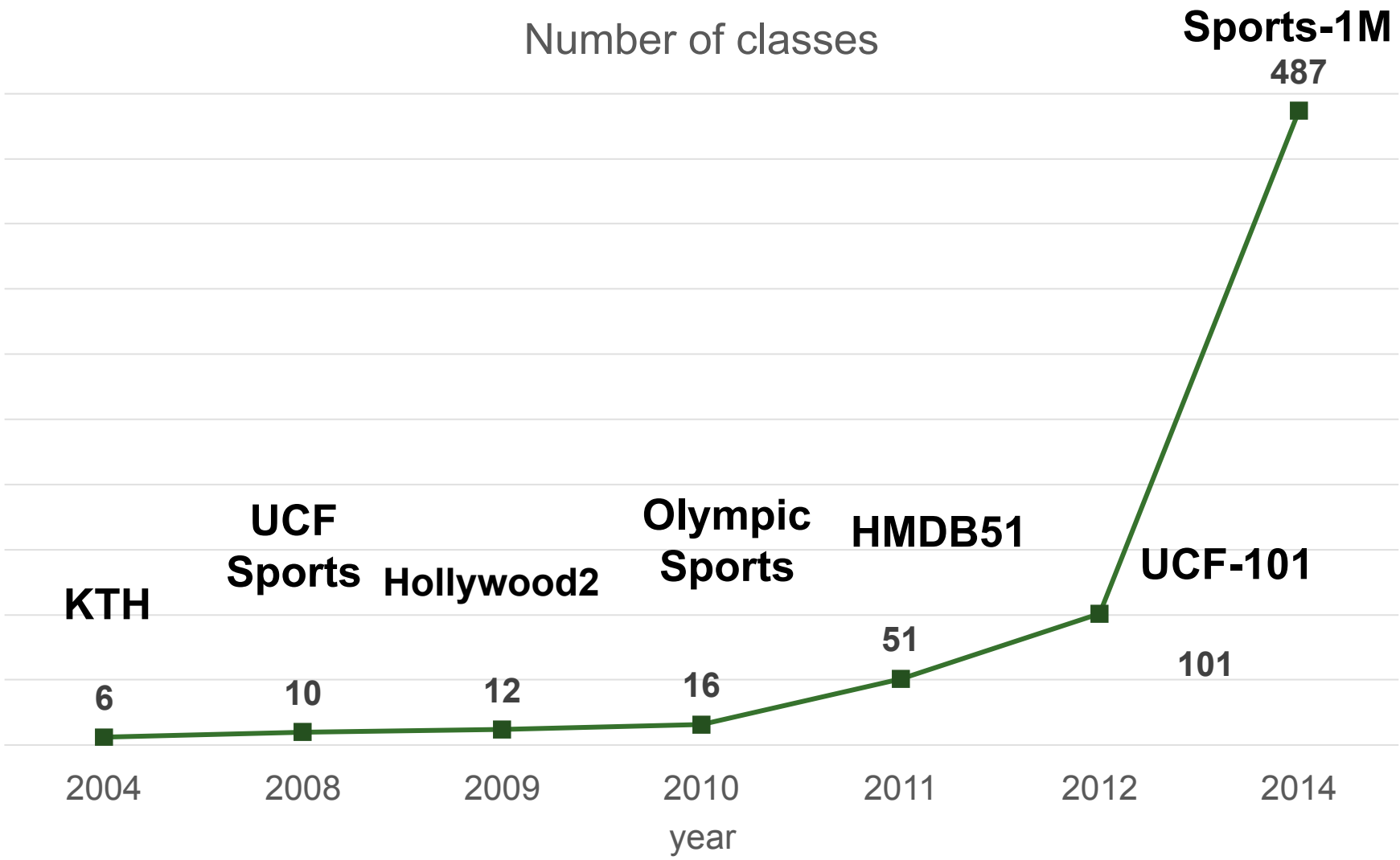
- Execution variability

- Camera angle variability

# Action Recognition - Challenges

- Occlusions, visual noise, shadows, different scales

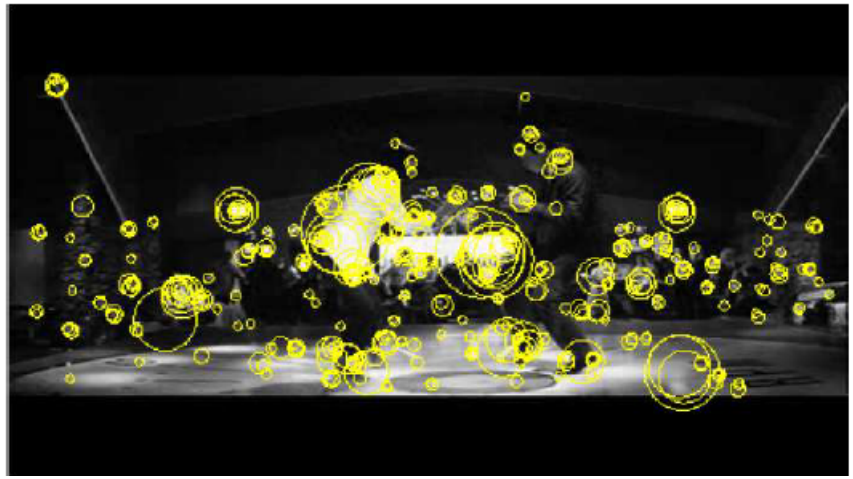
# Action Recognition - Datasets Evolution



# Video Processing and Action Recognition using Local Representations

# Local Representations for Visual Processing

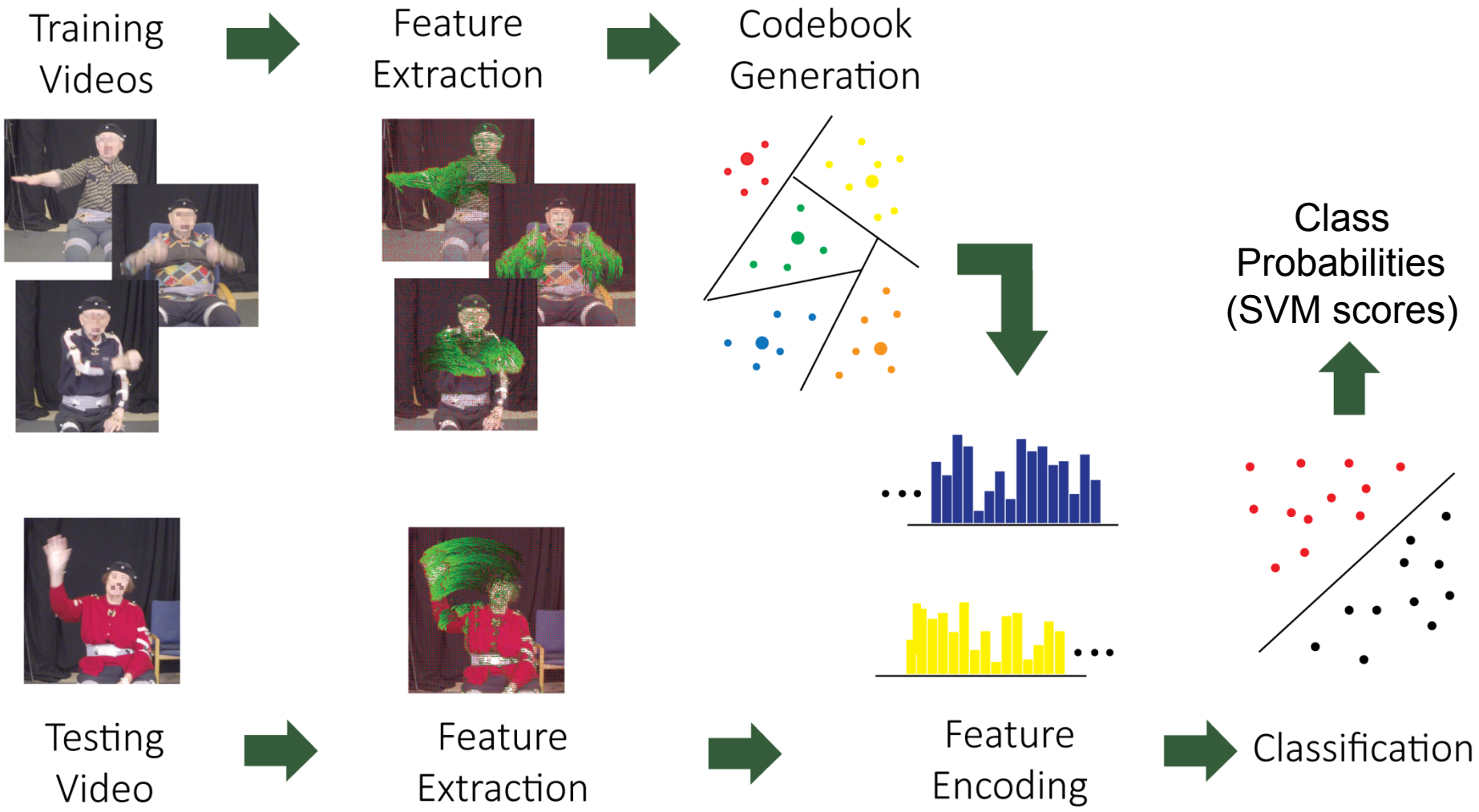
- Local video representations
  - Describe the whole video as a set of independent local descriptors
  - Detect independent interest points according a saliency function
  - Describe the detected points with features descriptors
  - Represent the video by encoding the statistical properties of the local interest points



2D Harris Detector

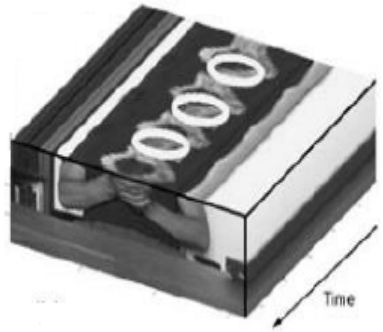
3D Harris Detector

# Traditional Action Recognition Pipeline



# Spatio-Temporal Interest Points (STIP)

- Detect points with large variation w.r.t. 3 direction of videos (x,y,t)
- Extract descriptors inside spatio-temporal volume around each interest point




**Descriptors:**

- **HOG:** static appearance (image gradient)
- **HOF:** motion (optical flow)

second moments matrix

$$M = g(x, y, t; \sigma_i^2, \tau_i^2) * \begin{pmatrix} L_x^2 & L_x L_y & L_x Lt \\ L_x L_y & L_y^2 & L_y L_t \\ L_x L_t & L_y L_t & L_t^2 \end{pmatrix}$$

Gaussian kernel

image gradient

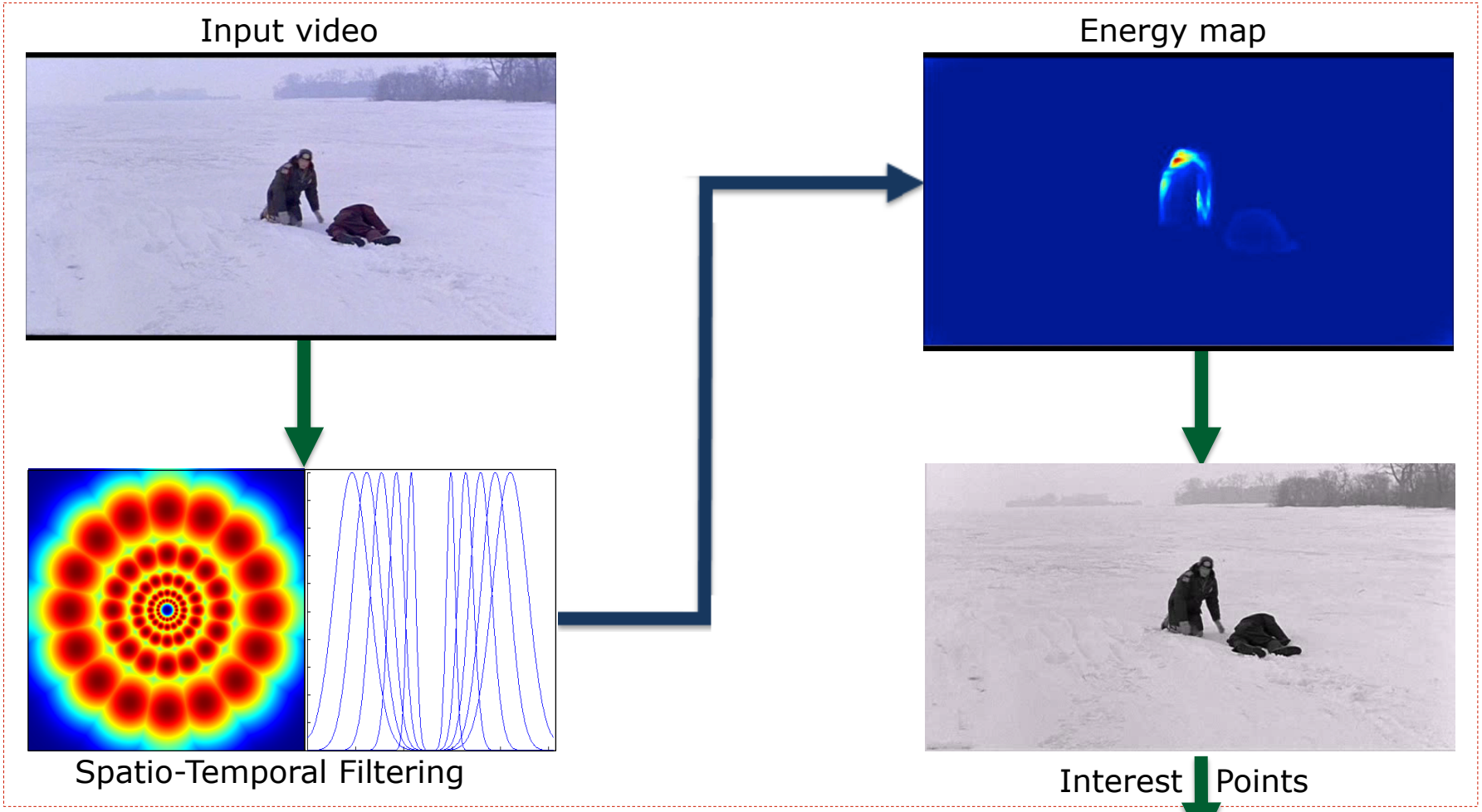
cornerness citerion

$$H = \underbrace{\lambda_1 \lambda_2 \lambda_3}_{\text{eigenvalues}} - k\left(\lambda_1 + \lambda_2 \lambda_3\right)^3$$

I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Proc. IEEE Conf. CVPR, 2008

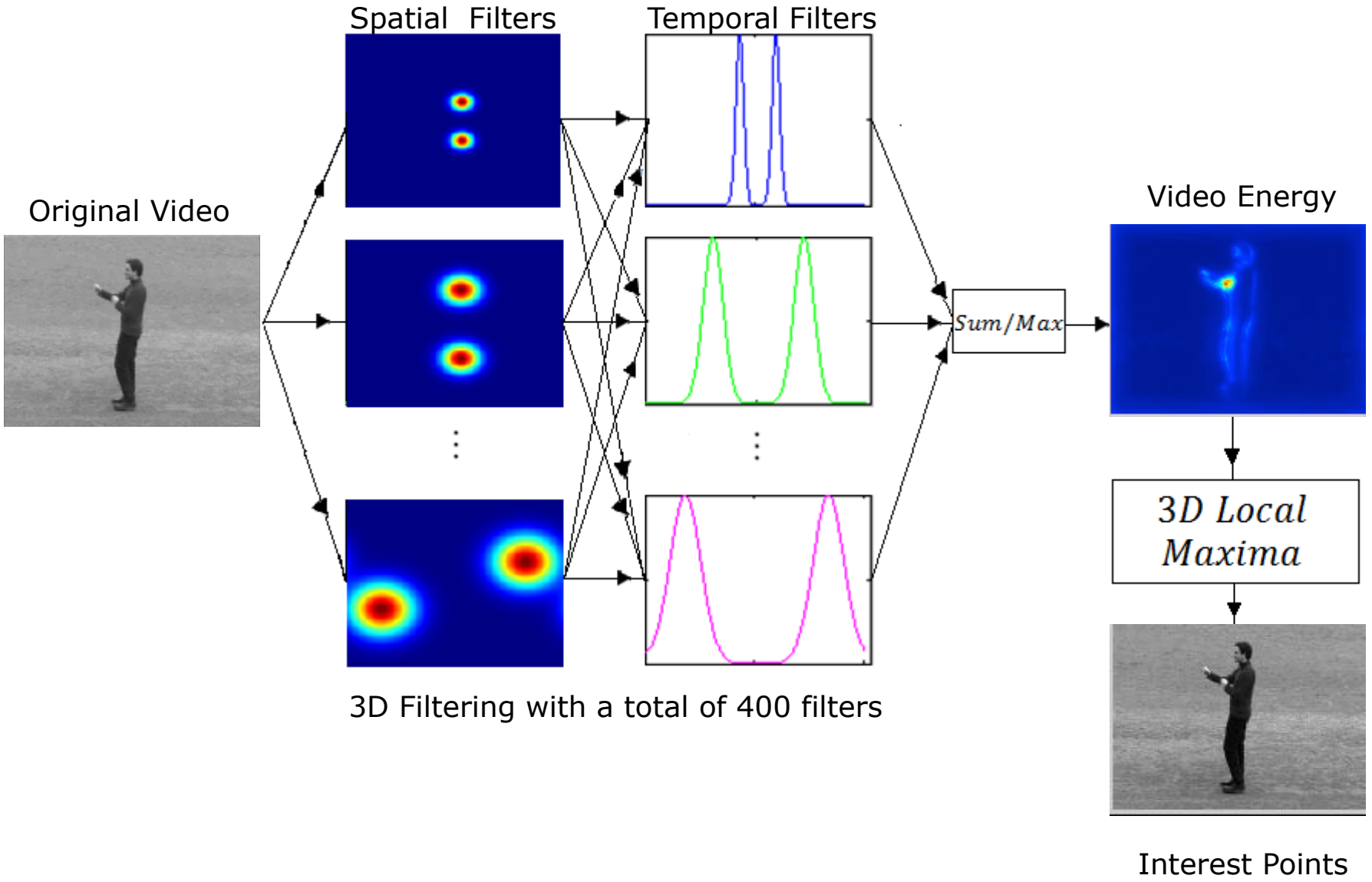
# Energy-based Interest Points

**Action Detection**




K. Maninis, P. Koutras and P. Maragos. Proc. ICIP 2014

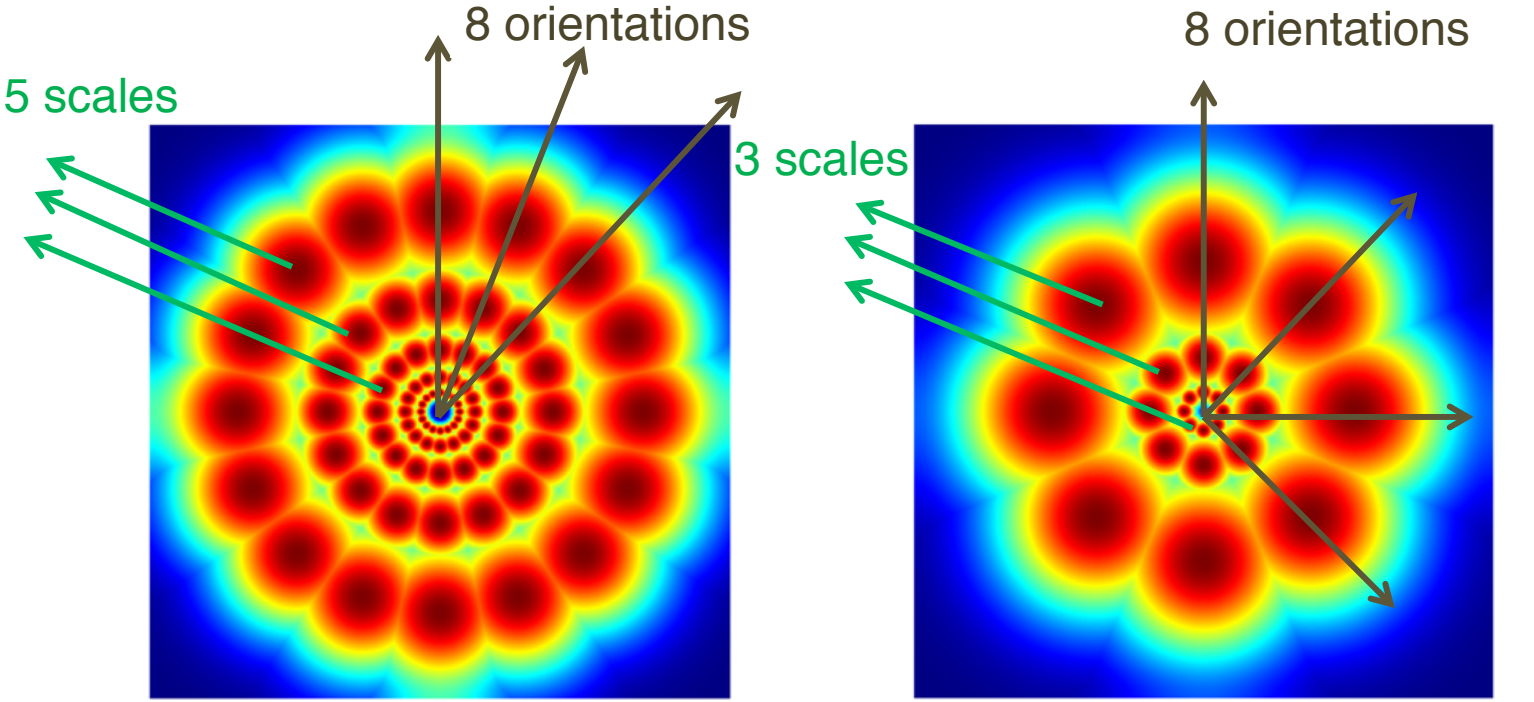
# Video processing with Gabor3D filterbanks



# Spatial Gabor Filterbank

Full Spatial Filterbank (40 Filters)

Reduced Spatial Filterbank (12 Filters)



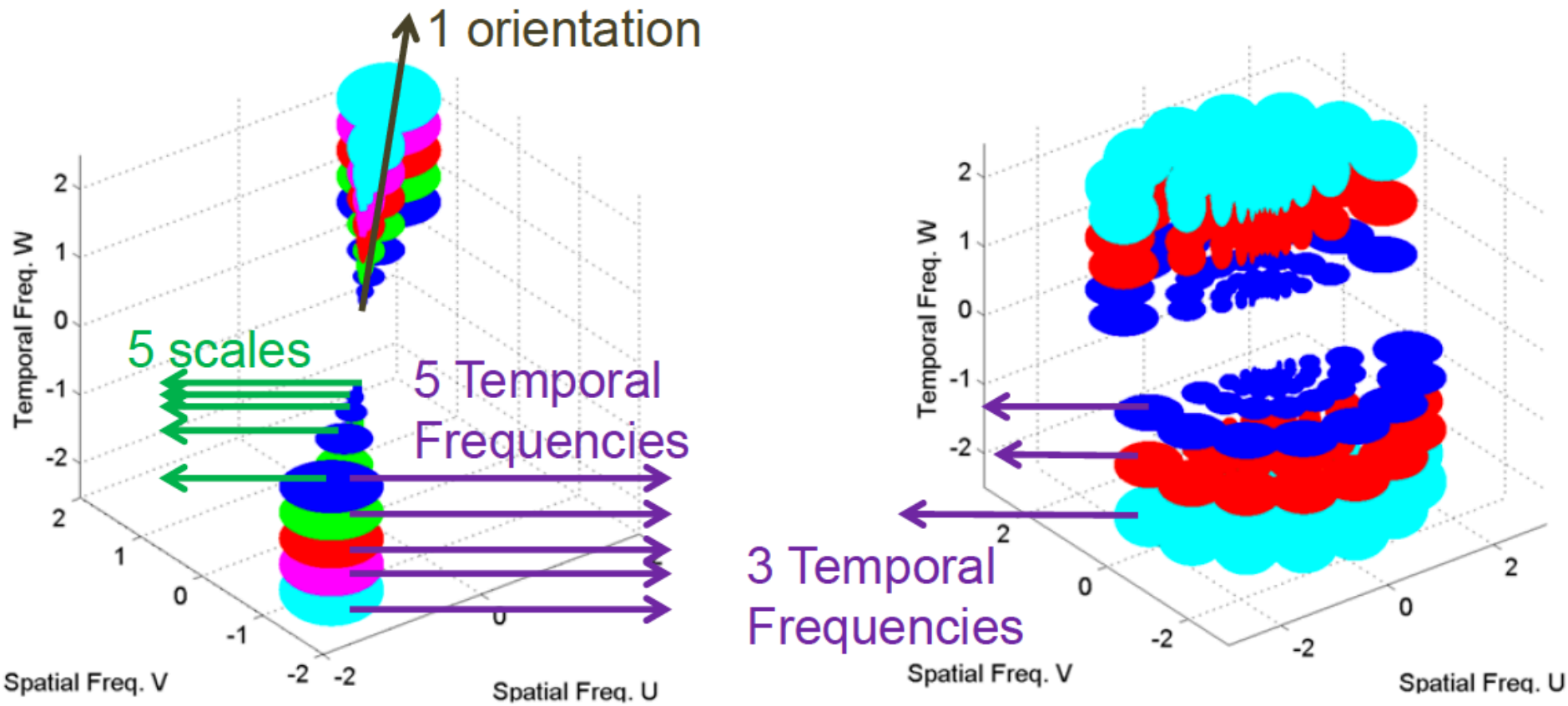

(a) Full filterbank at $\omega_{t_0}$ and $-\omega_{t_0}$ (top view).

(b) Reduced filterbank at $\omega_{t_0}$ and $-\omega_{t_0}$ (top view).

5 Positive & 5 Negative Temporal Frequencies

# Spatio-Temporal Gabor Filterbank

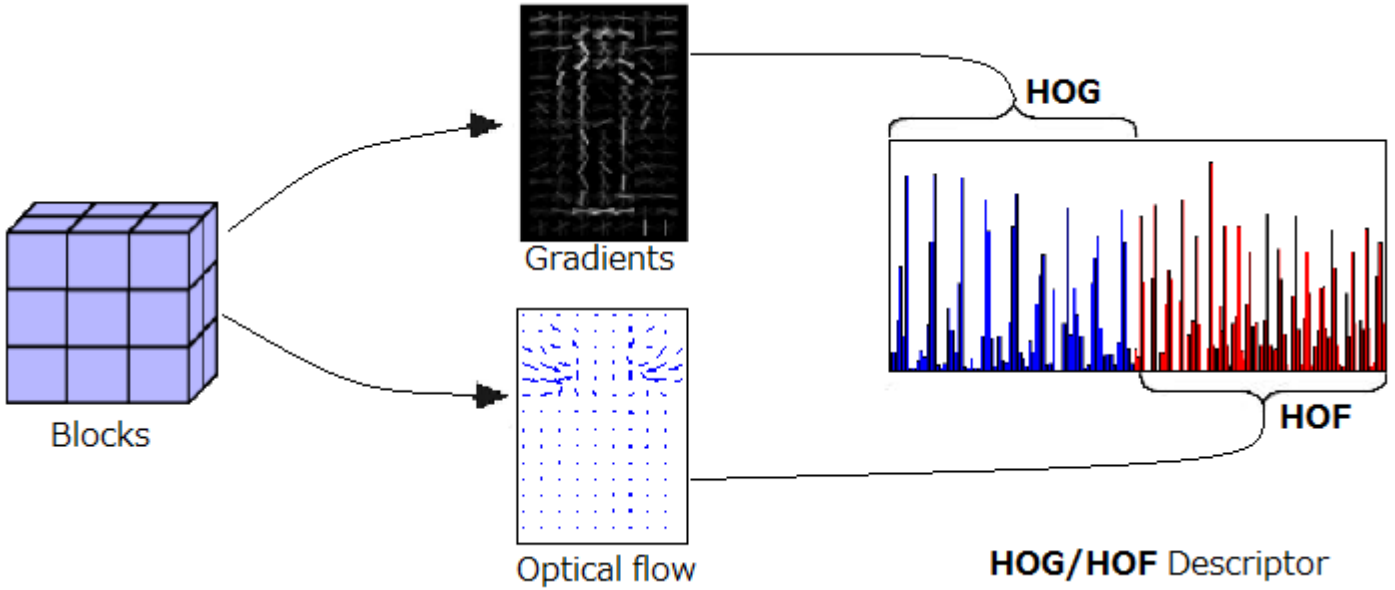
Full Spatio-Temporal Filterbank in 3D Space



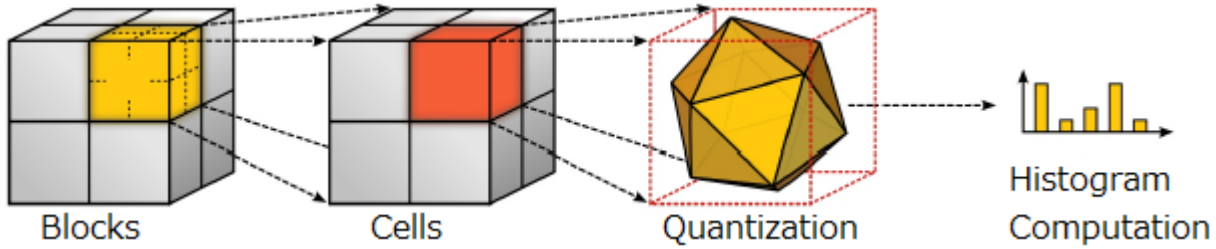

P. Koutras and P. Maragos. Signal Processing: Image Communication, vol 38, pp. 15-31, 2015.

# Histographic Gradient Descriptors

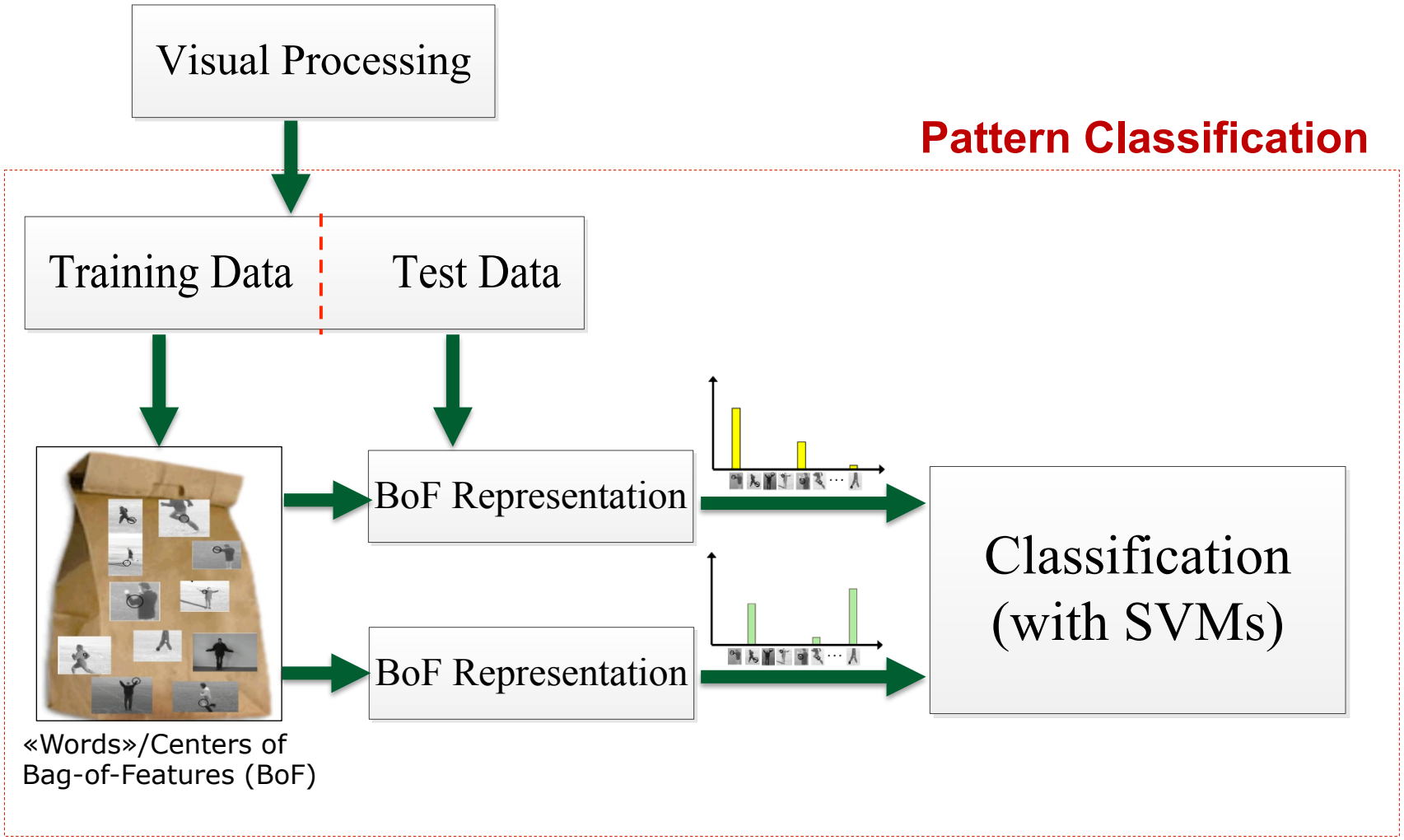
- HOG/HOF [Dalal et.al, CVPR 05, ECCV 06], [Laptev et.al, CVPR 08]




- HOG3D [Kläser et.al, BMVC 08]



# Visual Action Classification



# Action recognition with STIP- Results

**KTH** Action Database

- 6 Actions, 2391 videos

***Accuracy of various methods on the KTH Action Dataset***

| Method | DCA3D | Cuboids | Harris3D | Gabor3D |
|---|---|---|---|---|
| Accuracy | 78.8% | 90% | 91.8% | **93.5%** |

**Hollywood2** Action Database

- 12 Actions, 1707 videos
  - Large variation in action performances
  - Moving camera and scene changes
  - Multiple actors, background clutter and occlusion

***Mean Average Precision for the 12 action classes of the Hollywood2 Dataset***

| Method | Cuboids | Harris3D | Gabor3D |
|---|---|---|---|
| mAP | 46.2% | 45.2% | **47.7**% |

# Action recognition with STIP- Challenges

- Camera movement
- Generalization
- Scene changes → visual noise

SitDown

SitDown

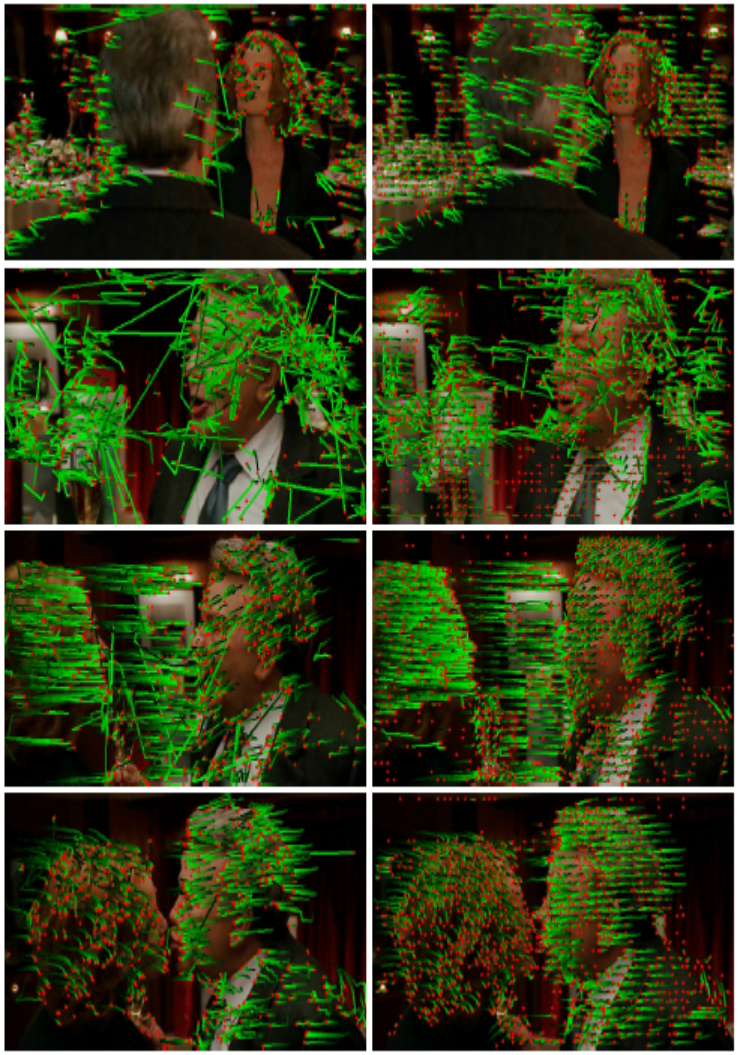
StandUp

StandUp

# Video Processing and Action Recognition using Dense Trajectories

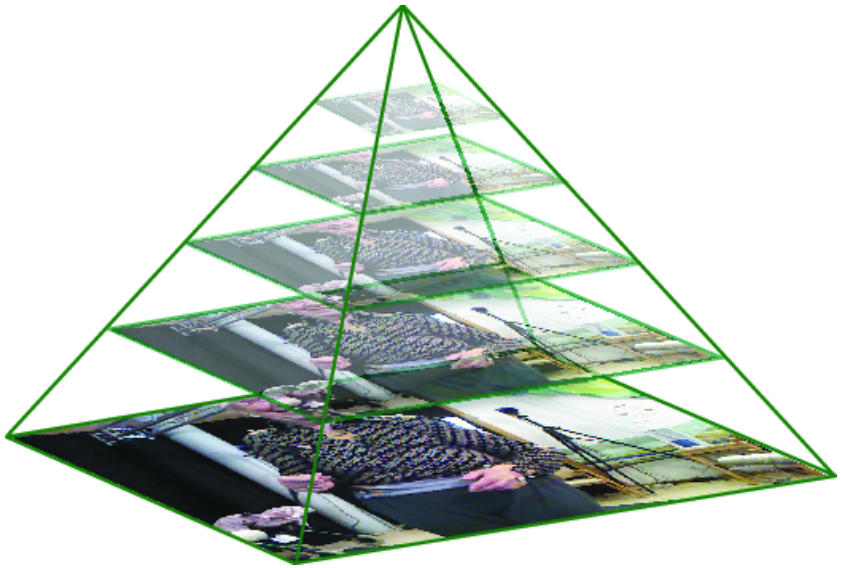
# Dense Trajectories - Overview

- Feature trajectories have shown to be efficient for representing videos
- The trajectories are obtained by tracking **densely sampled** points rather than **sparse STIP** using optical flow fields
- A local descriptor is introduced that overcomes the problem of camera motion
- The descriptor extends the motion coding scheme based on motion boundaries

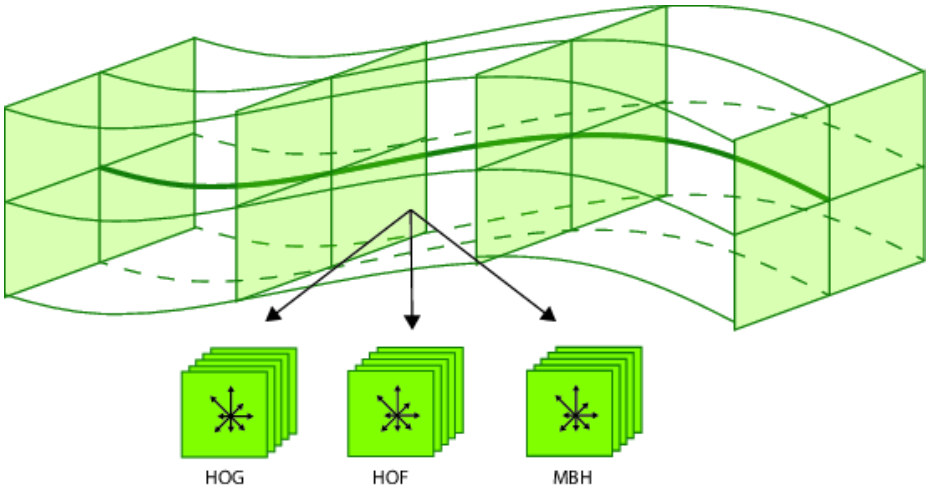



H. Wang, A. Klaser, C. Schmid, and C. Liu, Proc. CVPR 2011.

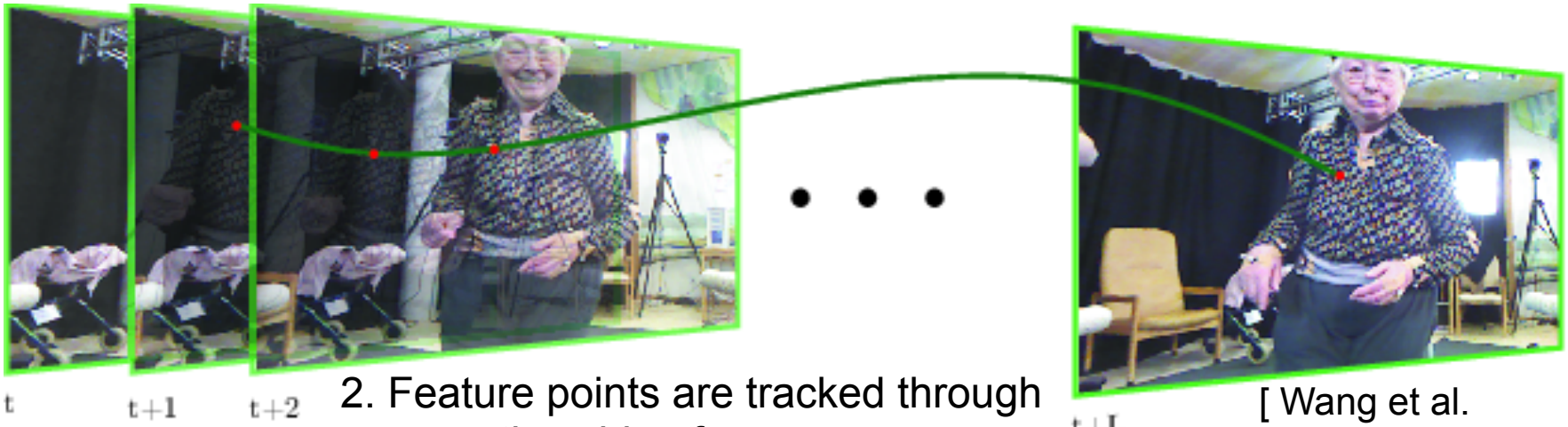
# Feature Extraction with Dense Trajectories



1. Feature points are sampled on a regular grid in multiple scales



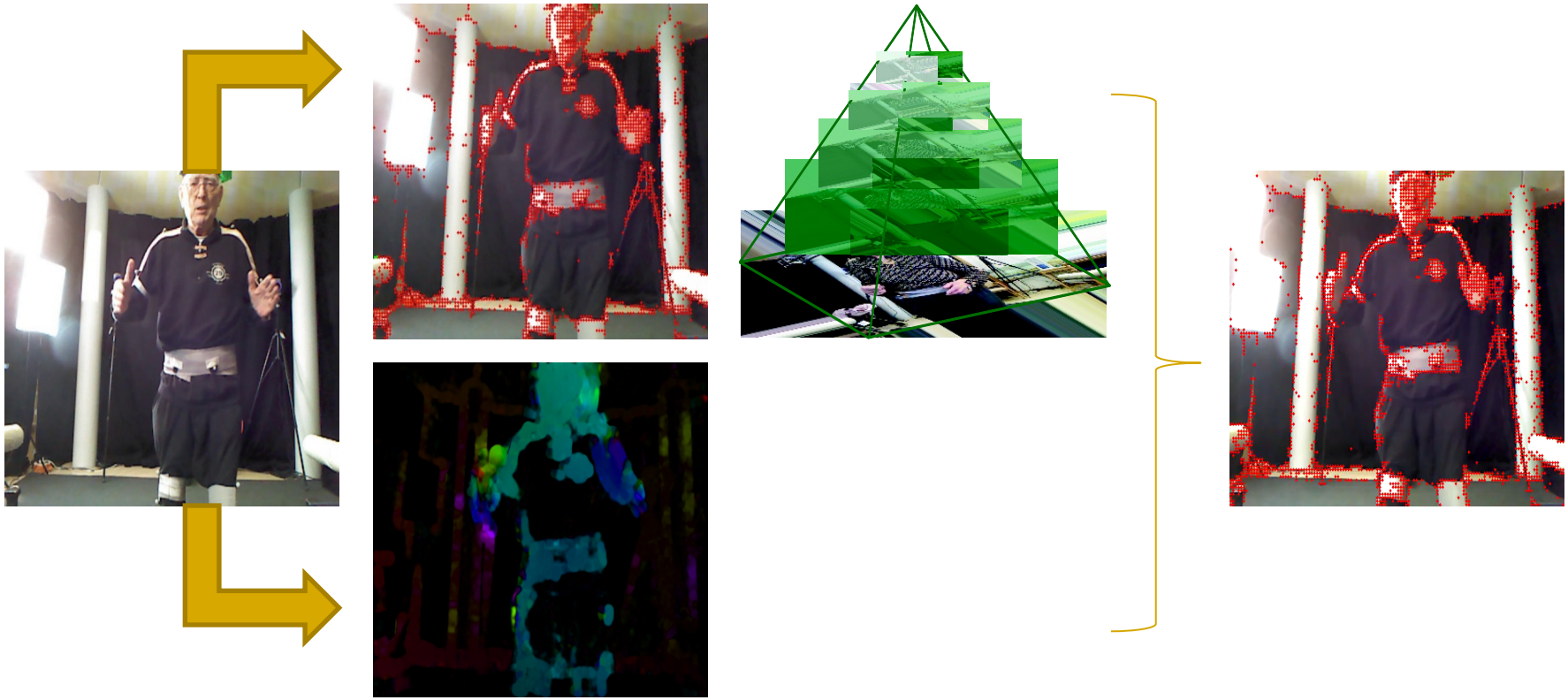

3. Descriptors are computed in space-time volumes along trajectories




2. Feature points are tracked through consecutive video frames

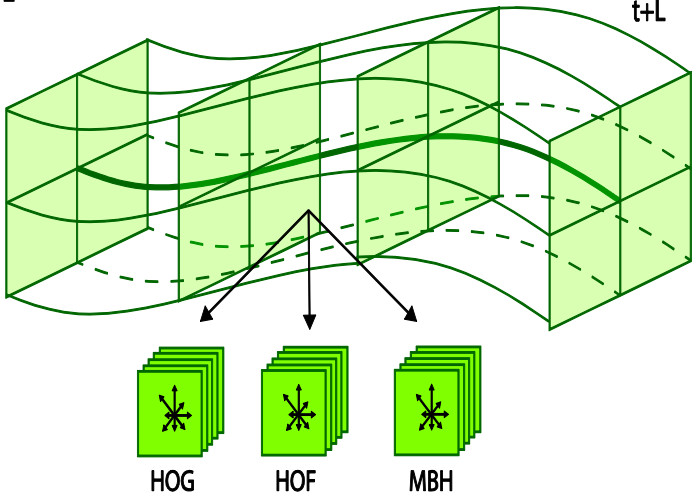
[ Wang et al. IJCV 2013 ]

# Dense Trajectories - Tracking



- Feature points are sampled on a grid spaced by W pixels and tracked in each scale separately (8 scales)
- Each point in a certain frame is tracked to the next frame using median filtering in a dense optical flow field

# Dense Trajectories - Descriptors

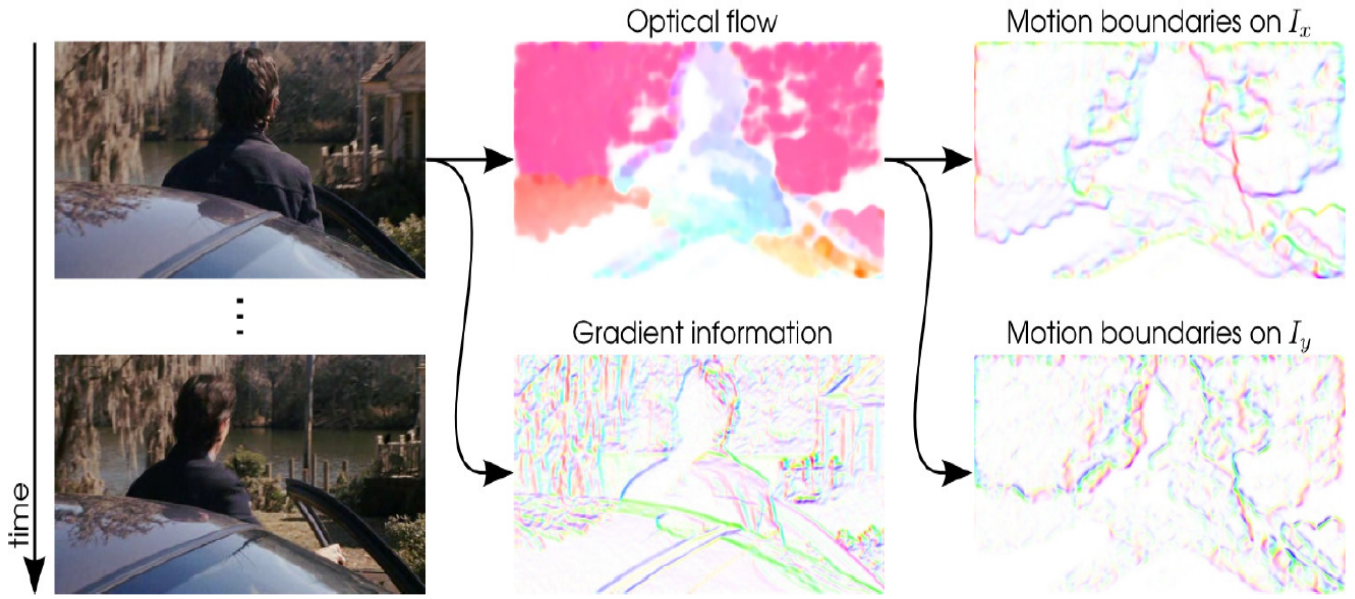

**Descriptors:**

- **HOG:** static appearance (image gradient)
- **HOF:** motion (optical flow)
- **MBH**: motion(motion gradient)
- **Trajectory:** consecutive points of the trajectory

- Trajectories are limited to 'L' frames in order to avoid drift from their initial location

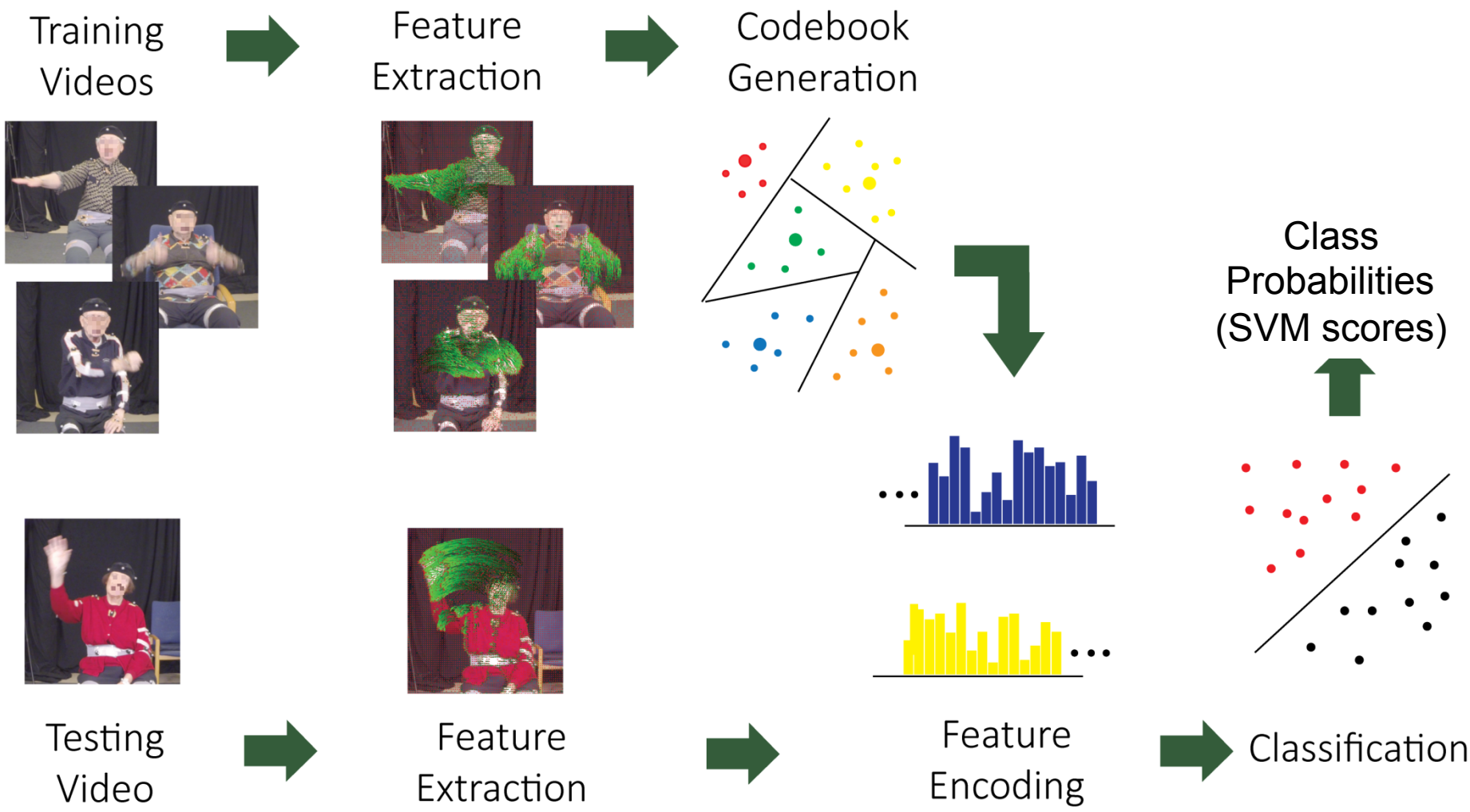
# Trajectory Descriptors

- Histogram of Oriented Gradient (HOG)
- Histogram of Optical Flow (HOF)
- HOGHOF
- Motion Boundary Histogram (MBH)
  - Take local gradients of x-y flow components and compute HOG as in static images

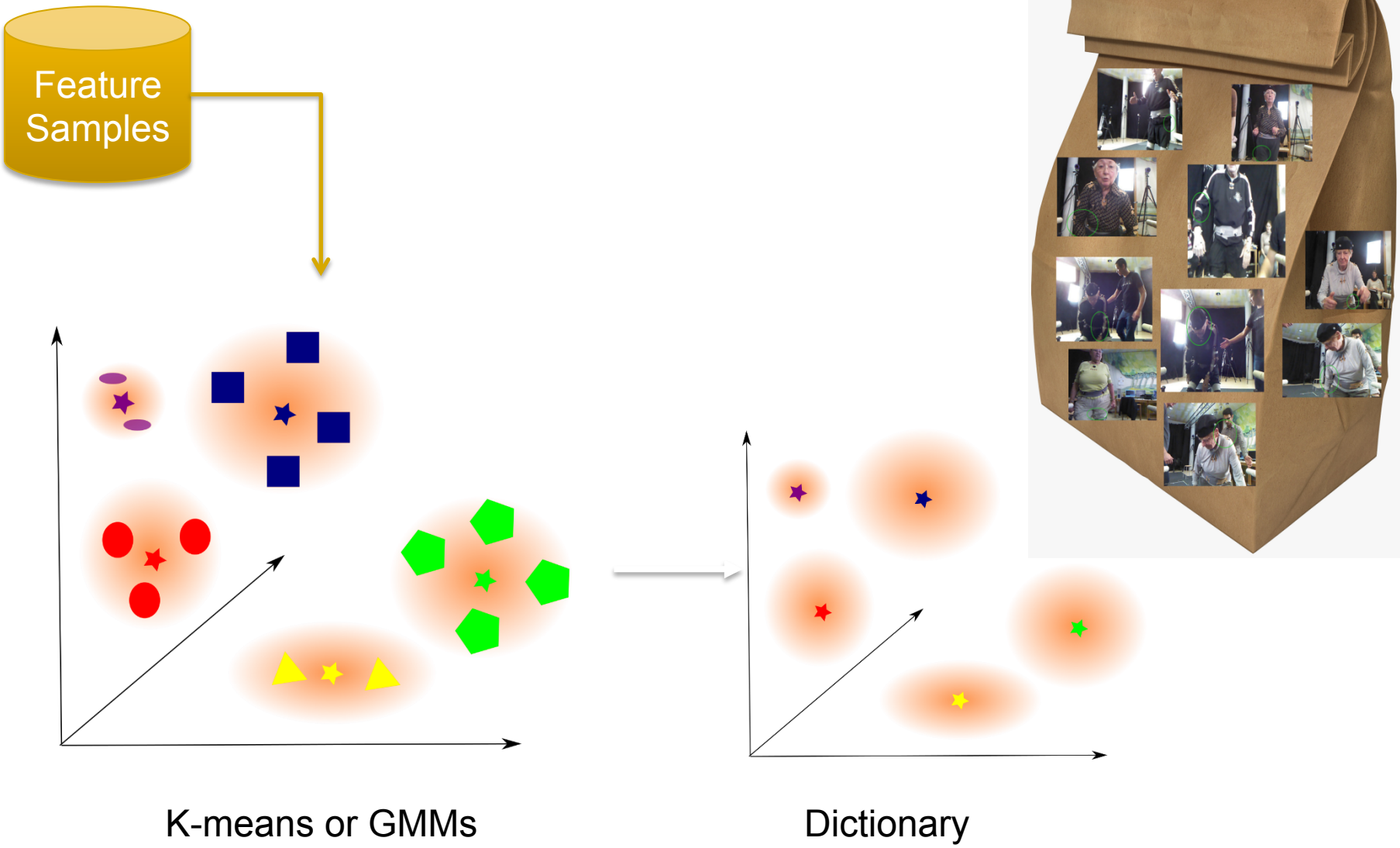



H. Wang, A. Klaser, C. Schmid, and C. Liu, Proc. CVPR 2011.

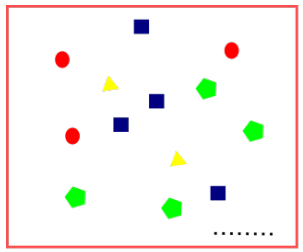
# Traditional Action Recognition Pipeline

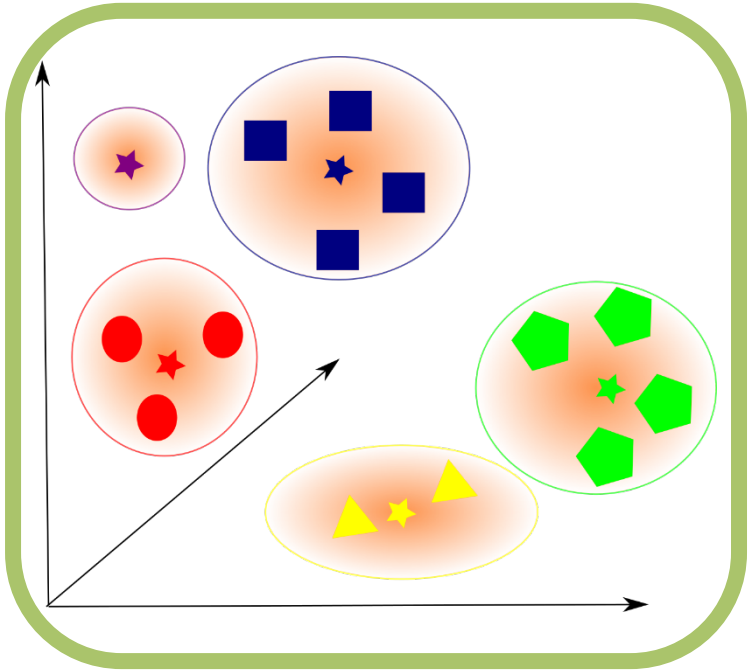


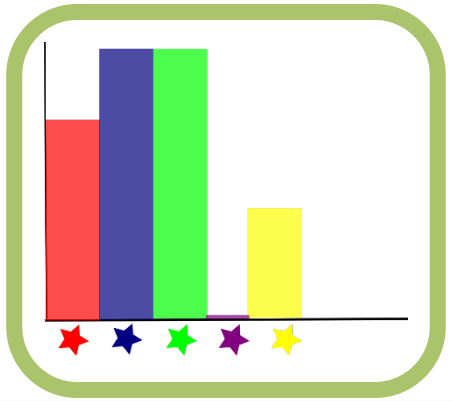
# Features Clustering and Dictionary



# Feature Encodings

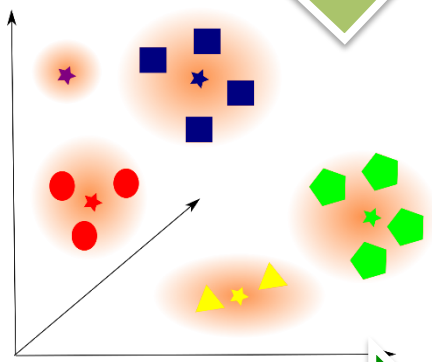




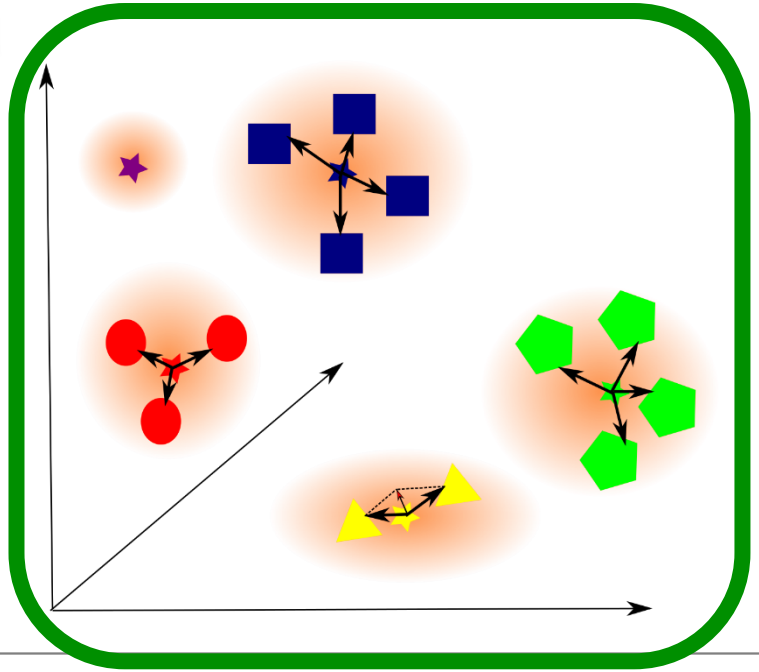


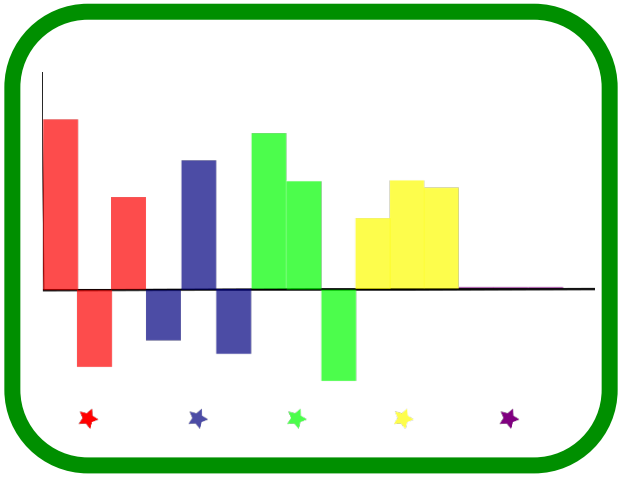
BOF - Size: K

BOF



VLAD





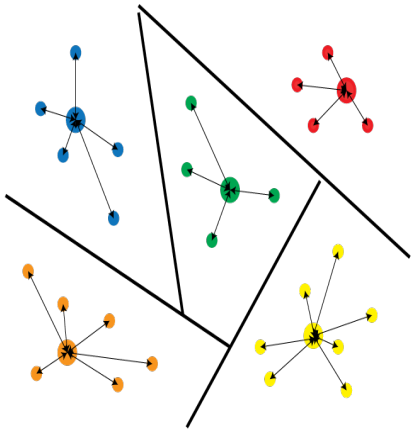
VLAD - Size: K*D

# Advanced Feature Encodings: VLAD, Fisher Vector

## VLAD:

$$\mathbf{v}_k = \sum_{\mathbf{x}_n \in \mathcal{C}_k} ||\mathbf{x}_n - \mathbf{d}_k||, k = 1 \dots K$$

VLAD: $\mathbf{v} = \begin{bmatrix} \mathbf{v}_1^T & \mathbf{v}_2^T & \cdots & \mathbf{v}_K^T \end{bmatrix}^T$



$\mathbf{d}_k$ : visual word

$\mathbf{x}_n$ : feature vector

## Fisher Vector:

$$\mathcal{G}_{\mu,k}^{\mathbf{X}} = \frac{1}{N\sqrt{w_k}} \sum_{n=1}^{N} \gamma_{n,k} \frac{\mathbf{x}_n - \mu_k}{\sigma_k}$$

$$\mathcal{G}_{\sigma,k}^{\mathbf{X}} = \frac{1}{N\sqrt{2w_k}} \sum_{n=1}^{N} \gamma_{n,k} \left( \frac{(\mathbf{x}_n - \mu_k)^2}{\sigma_k^2} - 1 \right)$$

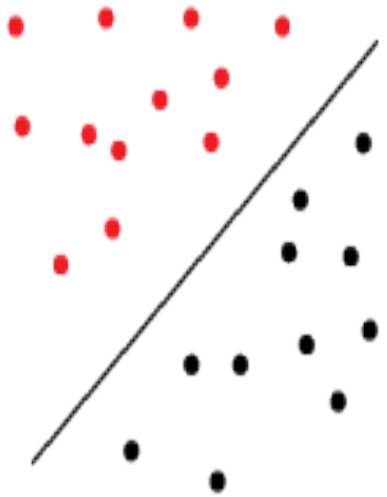
FV: $\left[ \left(\mathcal{G}_{\mu,1}^{\mathbf{X}}\right)^T \quad \left(\mathcal{G}_{\sigma,1}^{\mathbf{X}}\right)^T \quad \cdots \quad \left(\mathcal{G}_{\mu,K}^{\mathbf{X}}\right)^T \quad \left(\mathcal{G}_{\sigma,K}^{\mathbf{X}}\right)^T \right]^T$

$\gamma_{n,k}$ : soft assignment of each feature $\mathbf{x}_n \in \mathbf{X}, n = 1 \dots N$ to the k-th GMM's Gaussian with parameters $\mu_k, \sigma_k$

# Classification

- Support Vector Machines
- Kernels:
  - Linear: $K(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^T \mathbf{x}_j$
  - Chi-squared: $D(\mathbf{x}_i, \mathbf{x}_j) = \frac{1}{2} \sum_{k=1}^{K} \frac{(u_k - w_k)^2}{u_k + w_k}$ distance between histograms

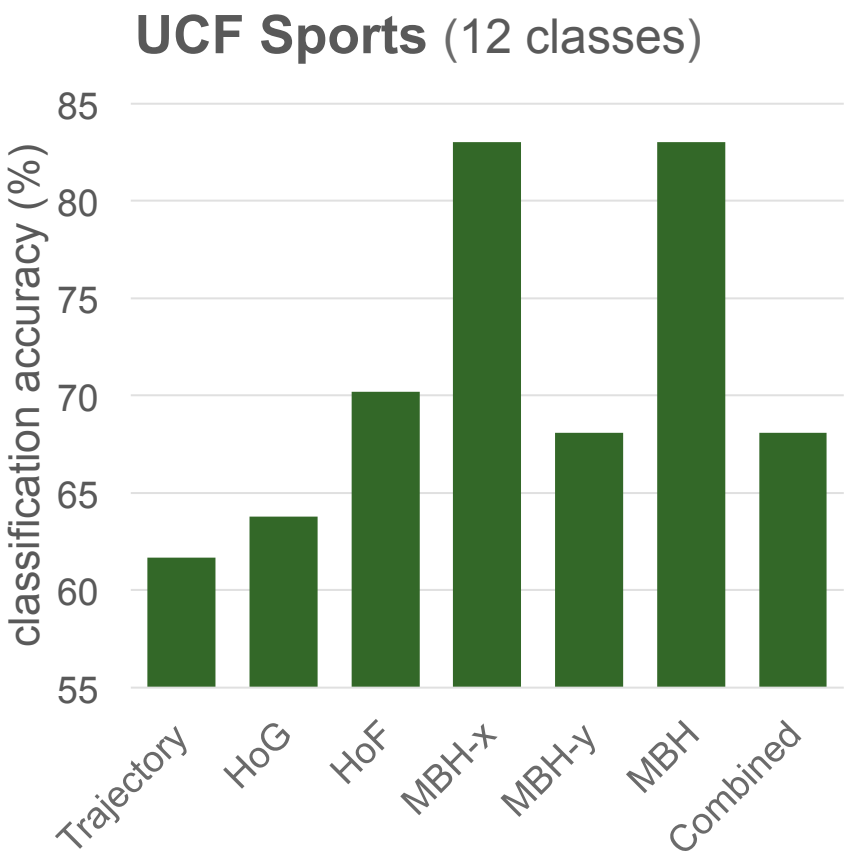
$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\frac{1}{A} D(\mathbf{x}_i, \mathbf{x}_j))$$ SVM kernel

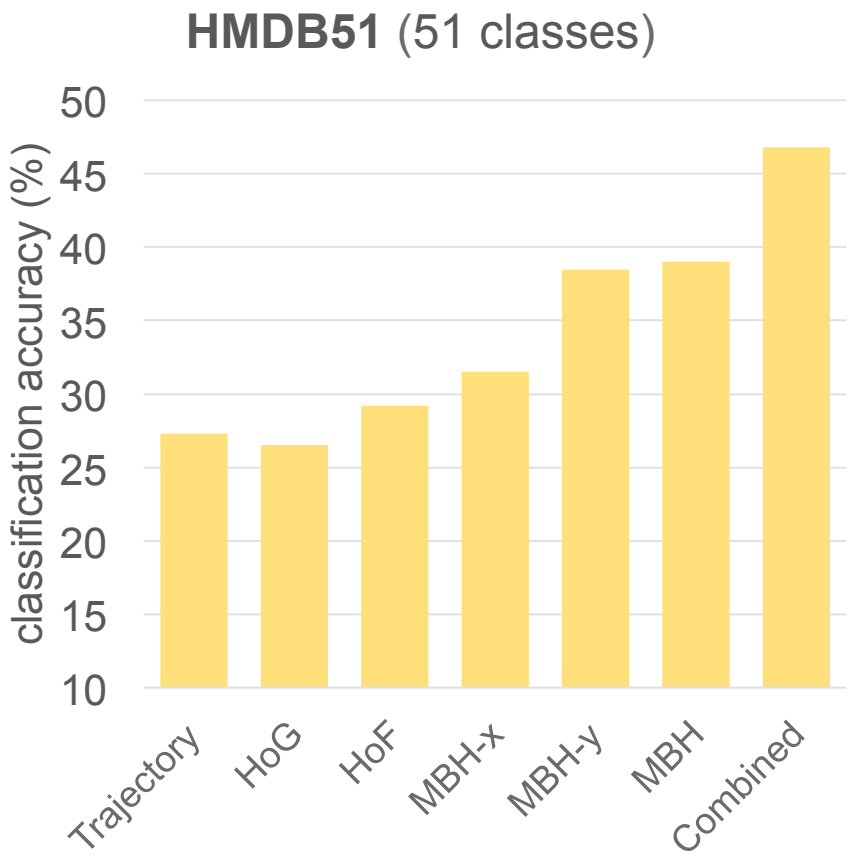


  - Features Fusion: $K(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\sum_c \frac{1}{A^c} D(\mathbf{x}_i^c, \mathbf{x}_j^c))$

c: different channels (descriptors)

# Dense Trajectories - Results

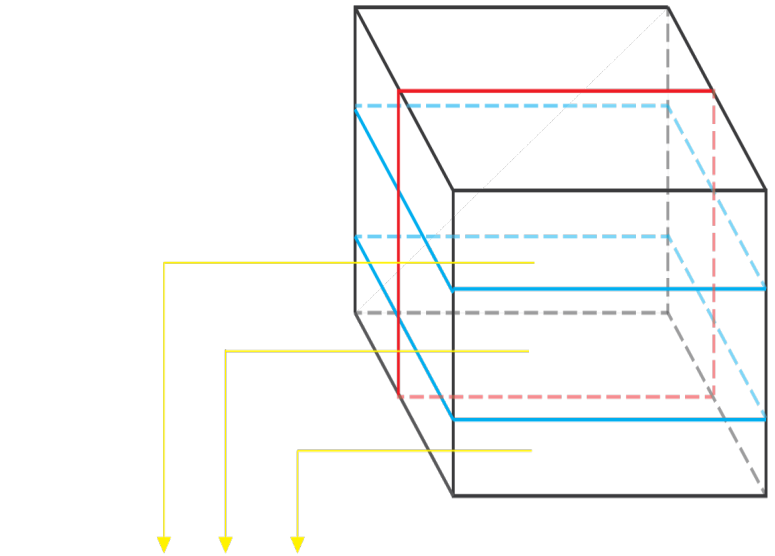


# Features Encoding: Spatio-Temporal Pyramids

**Integrate spatial and temporal structure:**

- Divide the 3D video volume into sub-volumes (cells)
- Compute a Bag-of-Words histogram for each cell
- Histogram concatenation

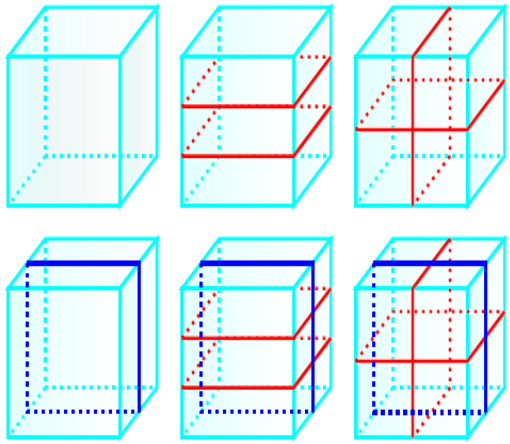


$$\mathbf{h} = [\mathbf{h}_1 | \mathbf{h}_2 | \mathbf{h}_3 | \ldots]$$

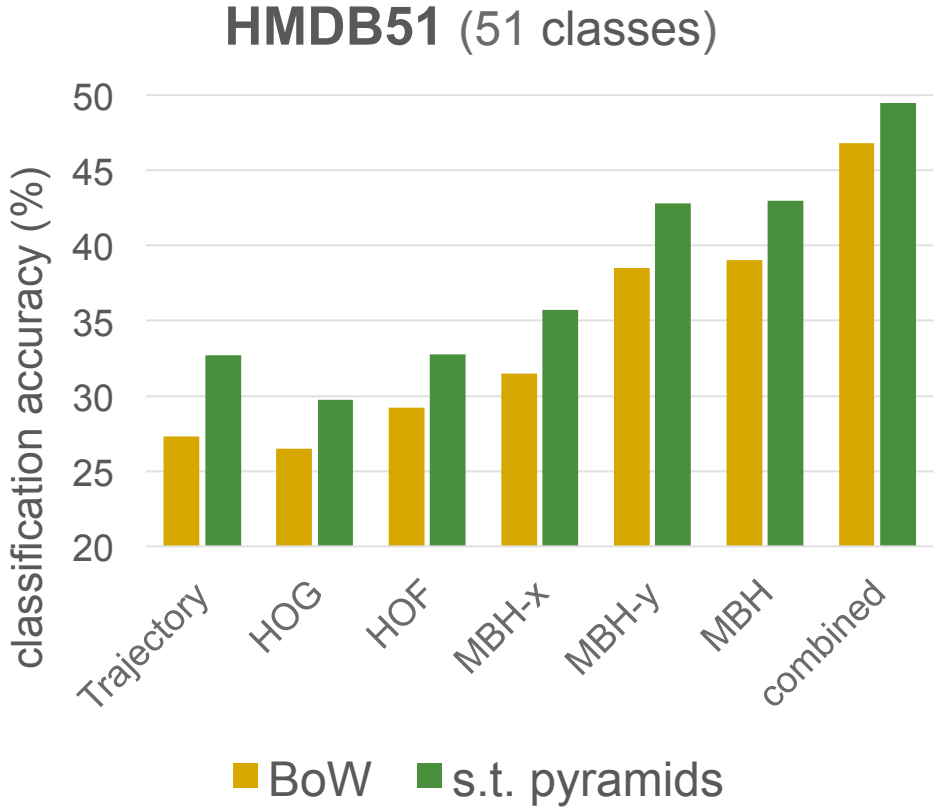
# Features Encoding: Spatio-Temporal Pyramids

- Channel fusion of different grids:

$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\sum_c \frac{1}{A^c} D(\mathbf{x}_i^c, \mathbf{x}_j^c))$$

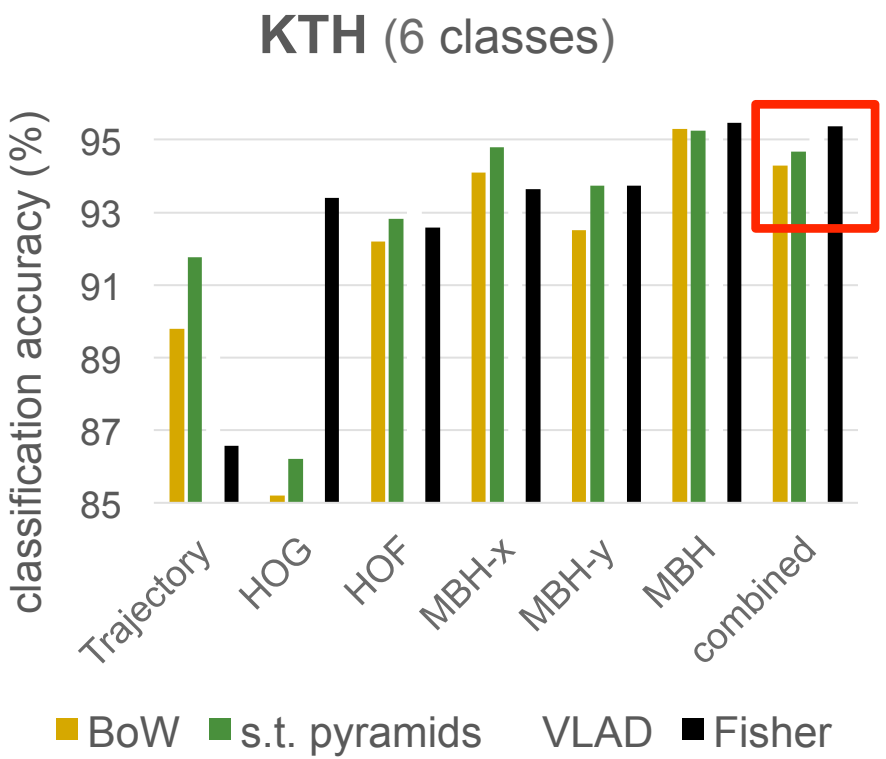


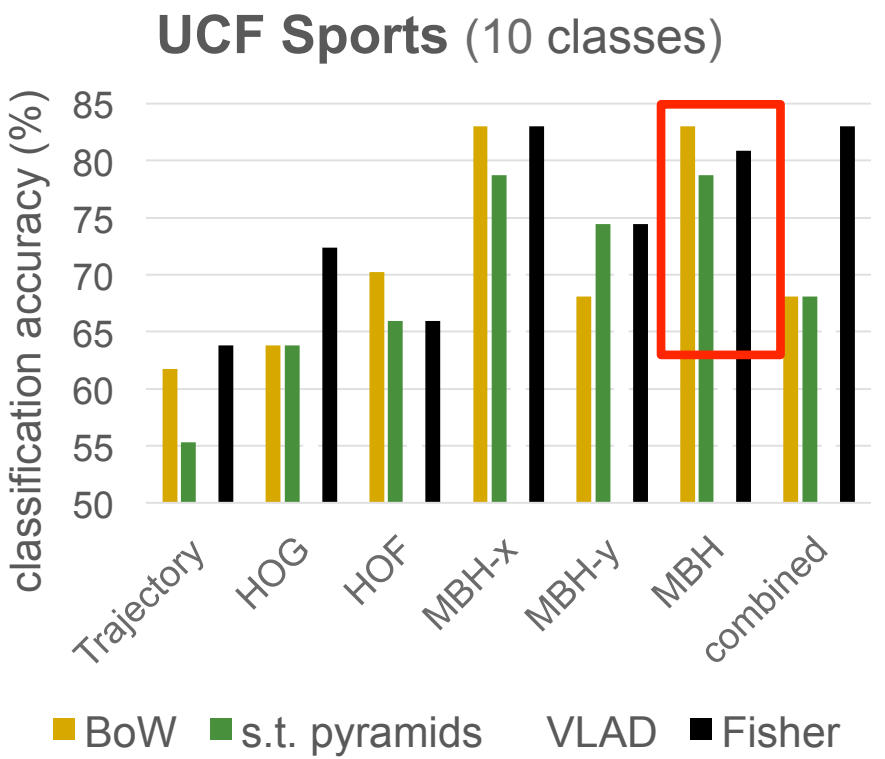
c: different channels (grids)



# VLAD and Fisher Vector: Results

- Better encoding depends on each problem/database

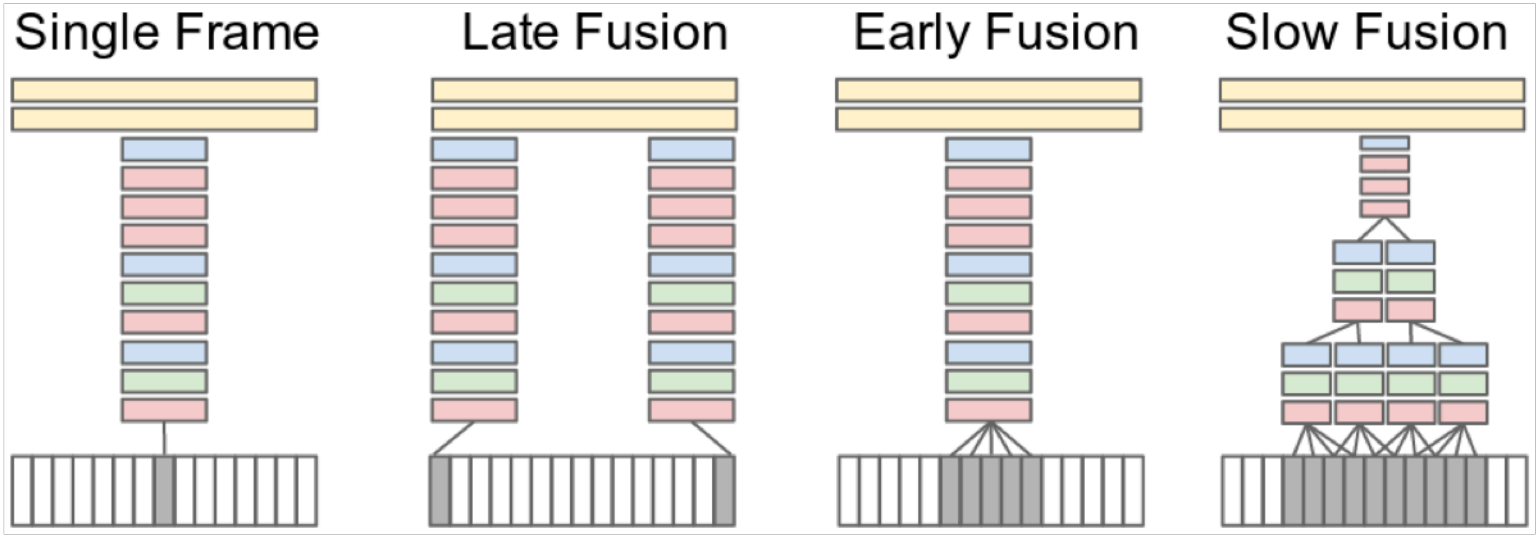


# Video Processing and Action Recognition using Deep Neural Networks

# Single Stream CNN-based Architectures

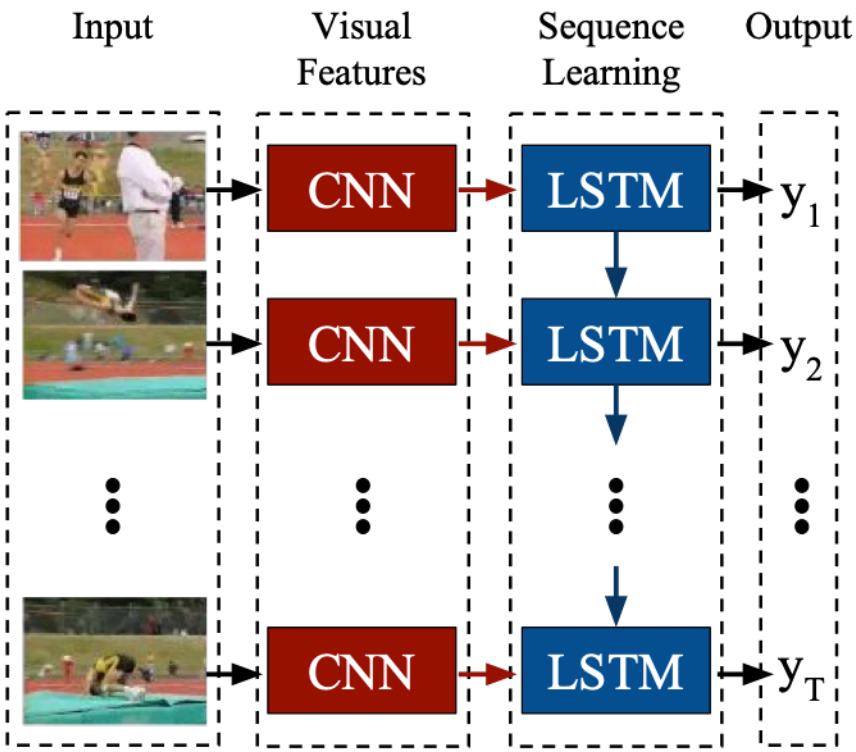
- Explore ways to fuse temporal information from consecutive frames using 2D pre-trained CNNs
- Cannot learn motion related patterns



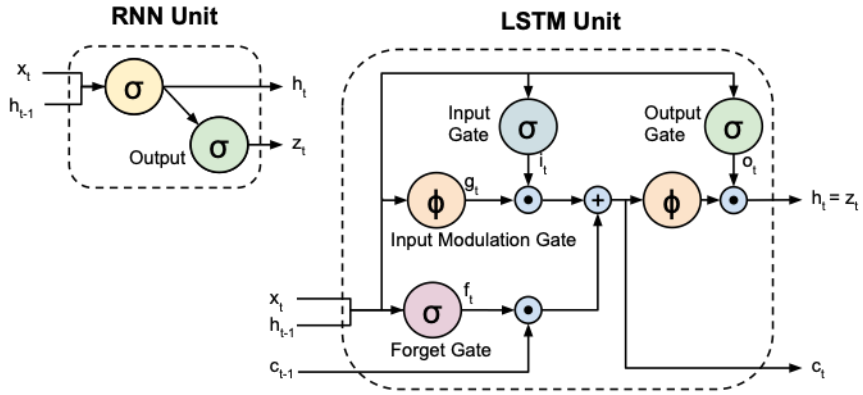

Karpathy et al. CVPR 2014.

# CNN + LSTM Architectures

- Long-term Recurrent Convolutional Networks
  - 2D CNN features (encoder)
  - employ LSTMs on top of them to capture the temporal information (decoder)
- end-to-end trainable architecture

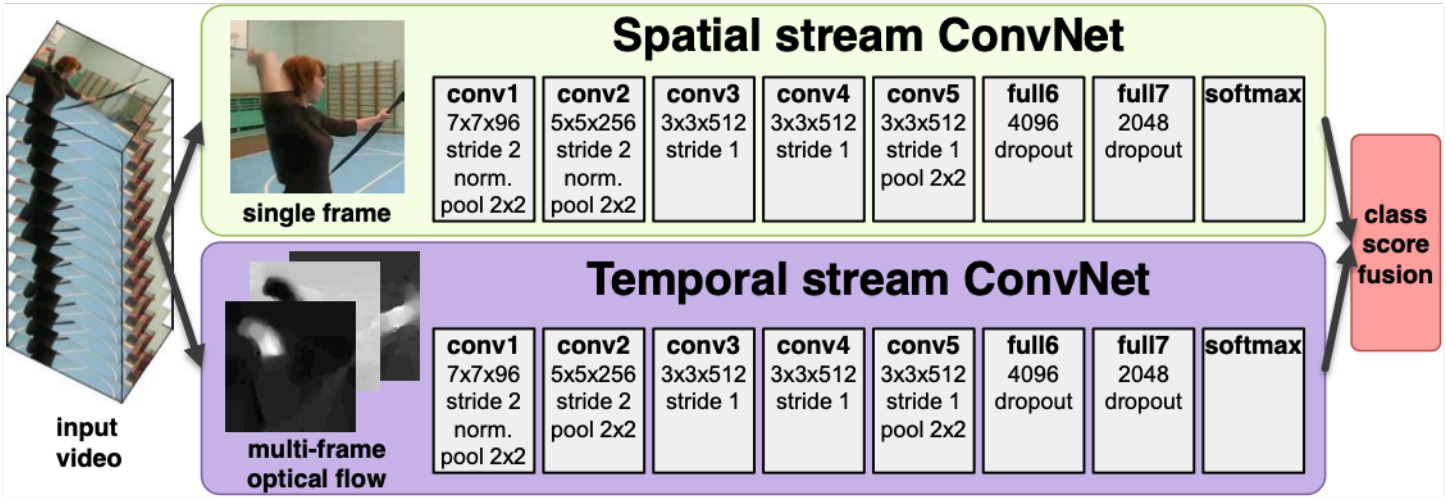



- RNN (Recurrent Neural Networks)
  - model temporal dynamics
- LSTM (Long Short Term Memory)
  - learn when to "forget" previous hidden states
  - learn when to update hidden states given new information




Donahue et al. Proc. CVPR, 2015

# Two Streams CNN: RGB + Optical Flow

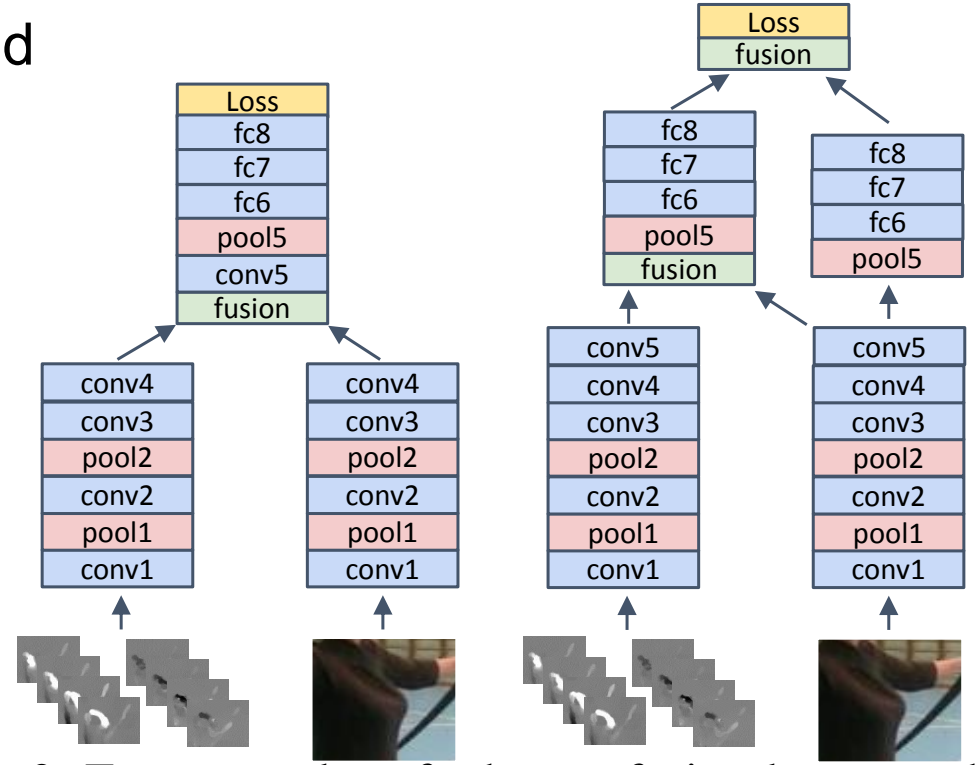



- 2D CNN architecture with two separate networks:
  - one for spatial context (pre-trained) → input: single video frame
  - one for motion context → optical flow stacked across for 10 successive frames
- explicitly capturing local temporal movement
- trained separately and combined using SVM

Simmoyan and Zisserman. Proc. NIPS, 2014

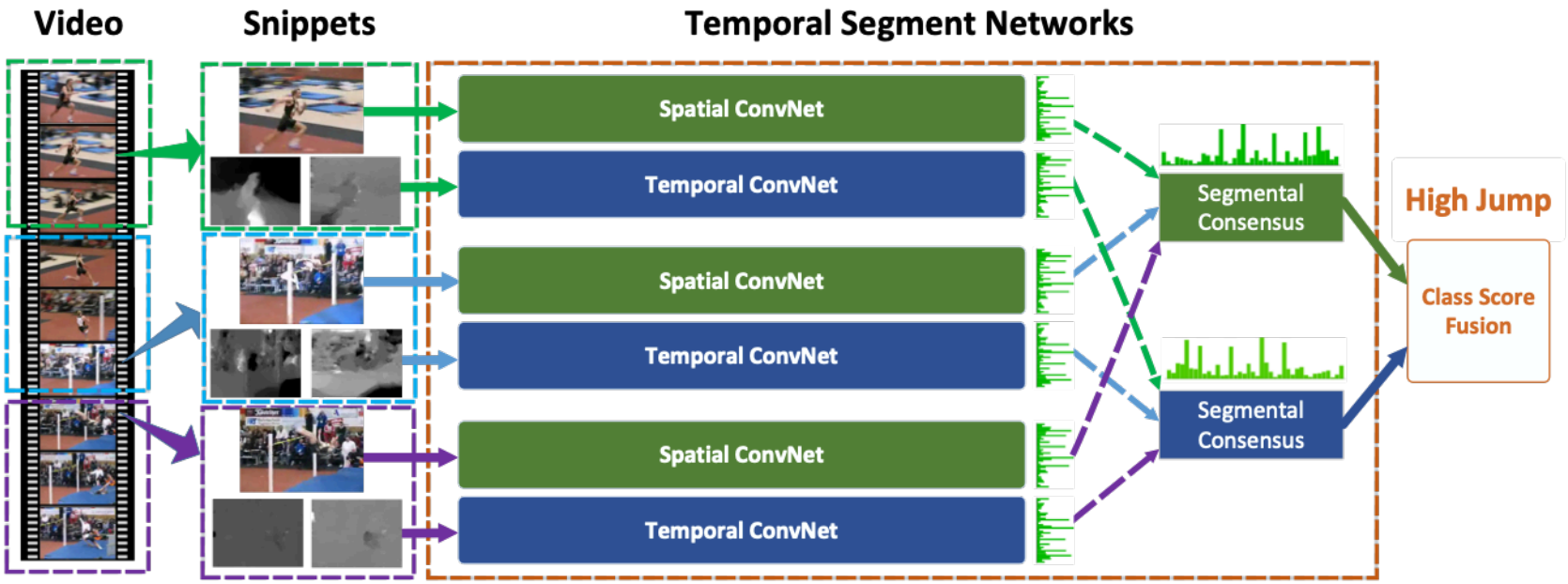
# Two Streams CNN: Advanced Fusion Schemes

- Fusion of spatial and temporal streams (how and when)
- Fusion at two layers (after conv5 and after fc8)
  - one as a hybrid spatiotemporal net
  - one as a purely spatial network
- Combining temporal net output across time frames → model long term dependencies




Feichtenhofer, Pinz and Zisserman. Proc. CVPR, 2016

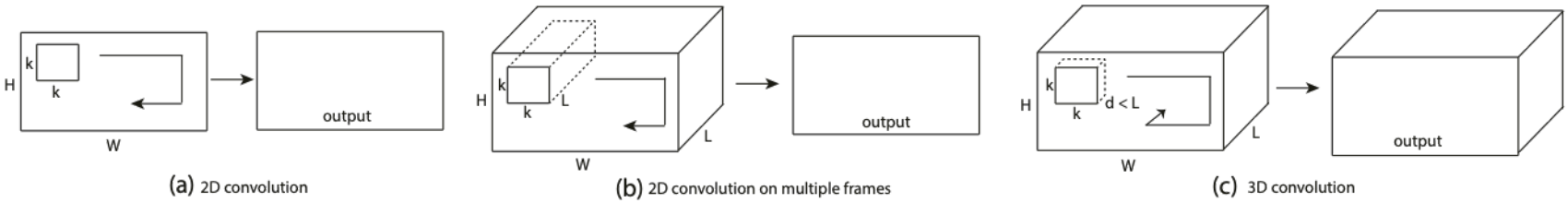
# Temporal Segment Networks



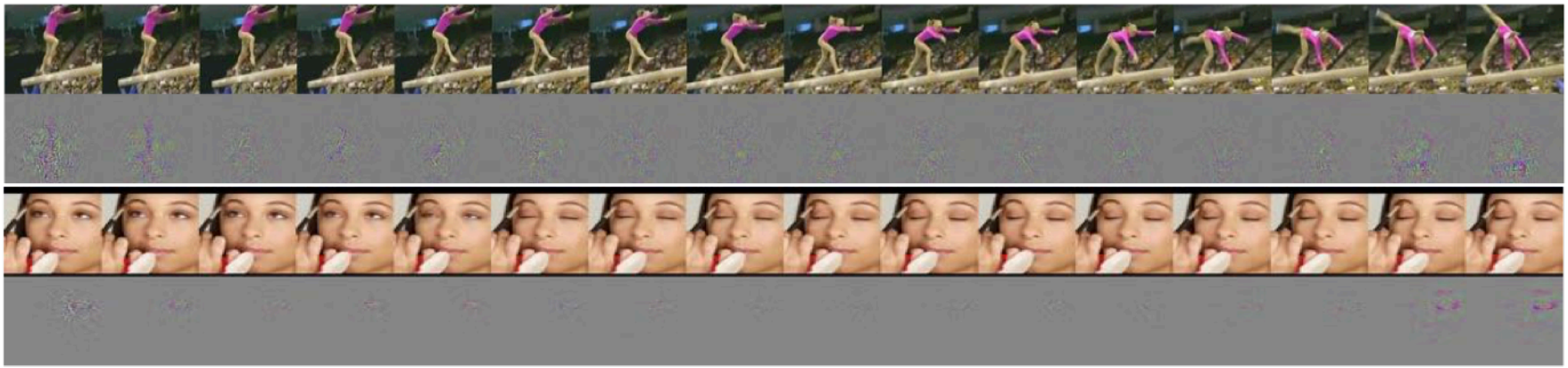

- two stream architecture
- suggest sampling clips sparsely across the video to better model long range temporal stimuli
- combine scores of temporal and spatial streams separately by averaging across snippets

L. Wang et al. Proc. ECCV, 2016

# 3D Convolutional Neural Networks

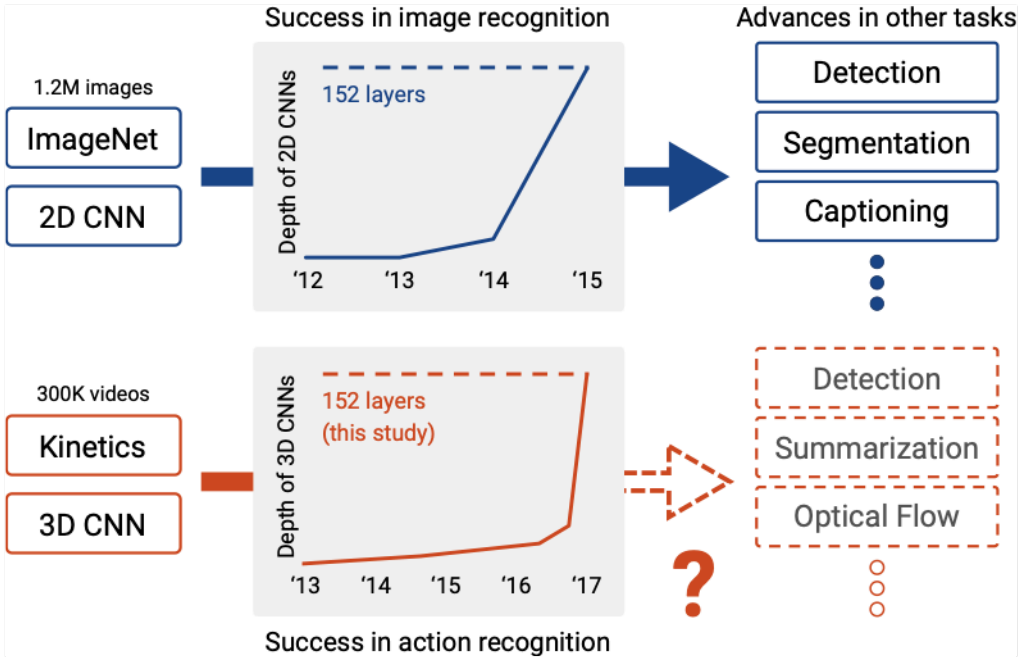


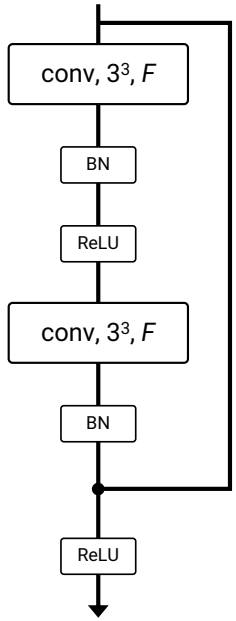

- Employ 3D convolutional networks as feature extractors instead of using 2D convolutions across frames
- Applying 3D convolution on a video volume results in another volume → preserving temporal information



D. Tran et al. Proc. ICCV, 2015

# Spatio-temporal 3D CNN – Transfer Learning



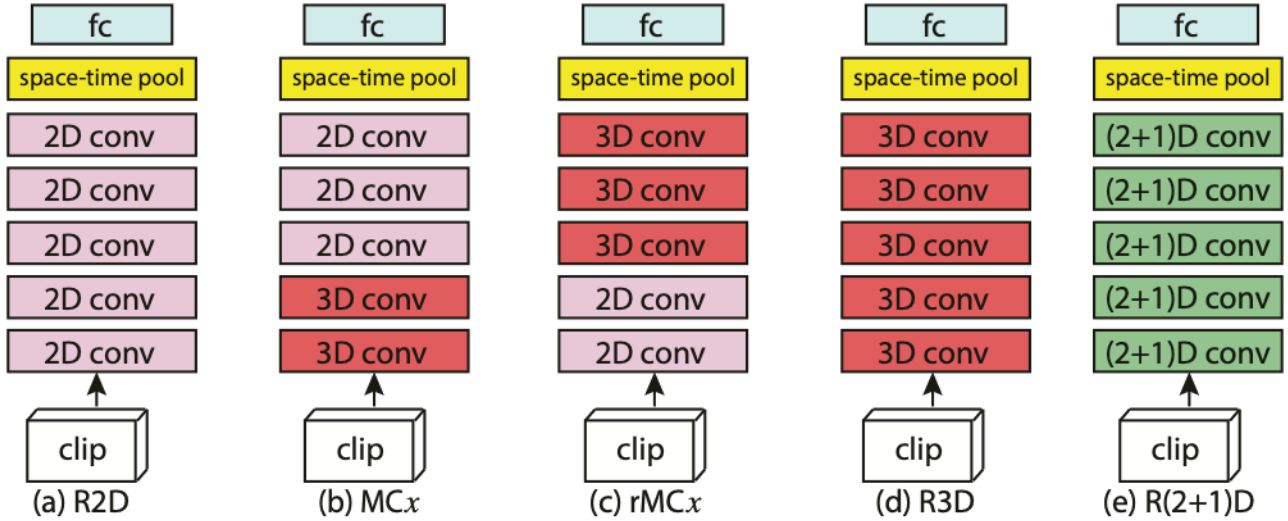



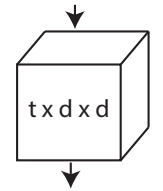

ResNet (basic)

- propose 3D CNNs based on ResNet architectures
- very deep 2D CNNs trained on ImageNet generates outstanding progress in image related tasks
- 3D CNNs trained on large scale action datasets can generate similar progress in computer vision for videos

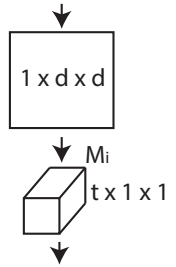
Hara, Kataoka and Satoh. Proc. CVPR, 2018

# Factorized Spatio-temporal CNN




3D conv




(2+1)D conv

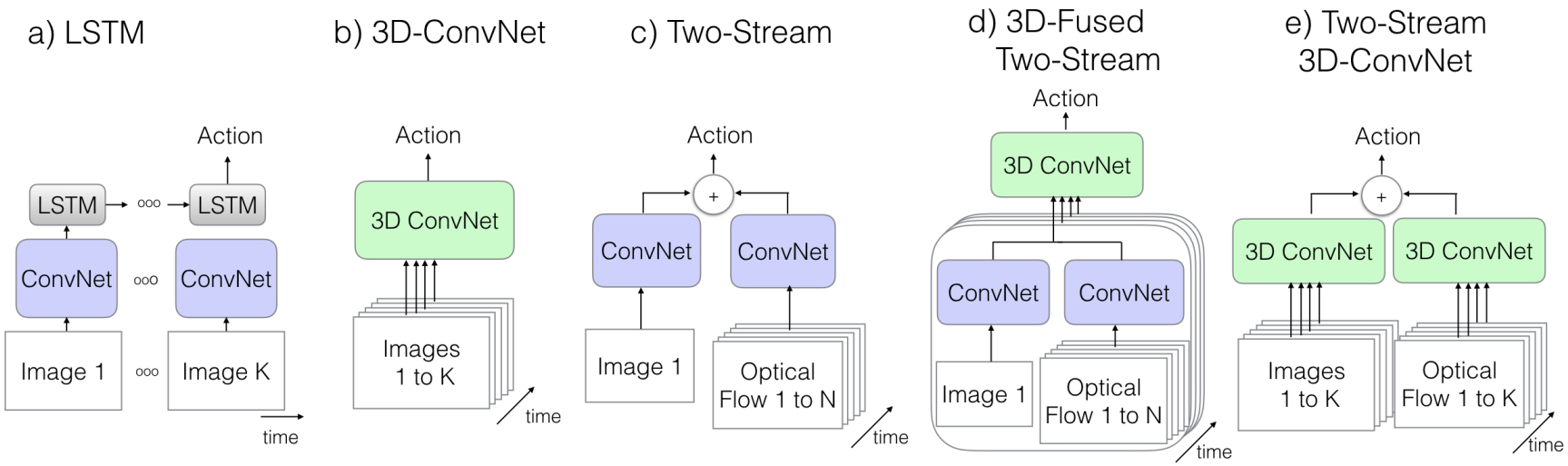
- empirically show that factorizing the 3D convolutional filters into separate spatial and temporal components yields significantly gains in accuracy
- design of a new spatiotemporal convolutional block “R(2+1)D”

| Net | # params | Clip@1 | Video@1 | Clip@1 | Video@1 |
|---|---|---|---|---|---|
| Input | | 8×112×112 | | 16×112×112 | |
| R2D | 11.4M | 46.7 | 59.5 | 47.0 | 58.9 |
| f-R2D | 11.4M | 48.1 | 59.4 | 50.3 | 60.5 |
| R3D | 33.4M | 49.4 | 61.8 | 52.5 | 64.2 |
| MC2 | 11.4M | 50.2 | 62.5 | 53.1 | 64.2 |
| MC3 | 11.7M | 50.7 | 62.9 | 53.7 | 64.7 |
| R(2+1)D | 33.3M | **52.8** | **64.8** | **56.8** | **68.0** |

D. Tran et al. Proc. CVPR, 2018

# Spatio-Temporal DNN-based Approaches

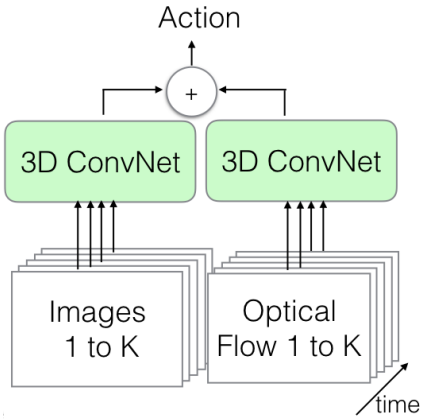
- Most DNN-based methods for video processing and recognition are improvisations on top of some basic approaches




Carreira and Zisserman, Proc. CVPR 2017

# I3D: Inflated 3D ConvNet

- Two-Stream Inflated 3D ConvNet (I3D) based on 2D ConvNet inflation
  - filters and pooling kernels of very deep image classification CNN are expanded into 3D
  - possible to learn seamless spatio-temporal feature extractors from video while leveraging successful ImageNet architecture designs/ parameters




| Model | UCF-101 | HMDB-51 |
|---|---|---|
| Two-Stream [27] | 88.0 | 59.4 |
| IDT [33] | 86.4 | 61.7 |
| Dynamic Image Networks + IDT [2] | 89.1 | 65.2 |
| TDD + IDT [34] | 91.5 | 65.9 |
| Two-Stream Fusion + IDT [8] | 93.5 | 69.2 |
| Temporal Segment Networks [35] | 94.2 | 69.4 |
| ST-ResNet + IDT [7] | 94.6 | 70.3 |
| Deep Networks [15], Sports 1M pre-training | 65.2 | - |
| C3D one network [31], Sports 1M pre-training | 82.3 | - |
| C3D ensemble [31], Sports 1M pre-training | 85.2 | - |
| C3D ensemble + IDT [31], Sports 1M pre-training | 90.1 | - |
| RGB-I3D, Imagenet+Kinetics pre-training | 95.6 | 74.8 |
| Flow-I3D, Imagenet+Kinetics pre-training | 96.7 | 77.1 |
| Two-Stream I3D, Imagenet+Kinetics pre-training | **98.0** | 80.7 |
| RGB-I3D, Kinetics pre-training | 95.1 | 74.3 |
| Flow-I3D, Kinetics pre-training | 96.5 | 77.3 |
| Two-Stream I3D, Kinetics pre-training | 97.8 | **80.9** |

Carreira and Zisserman, Proc. CVPR 2017

# Multi-task Spatio-Temporal Networks for Video Understanding

Koutras and Maragos, SUSiNet: “See, Understand and Summarize It”, Proc. CVPRW, 2019

# Action Recognition

- automated classification and detection of human activities on videos
  - action labels from human annotations
  - many large datasets: Hollywood2, UCF101, HMDB51, Kinetics

open car door

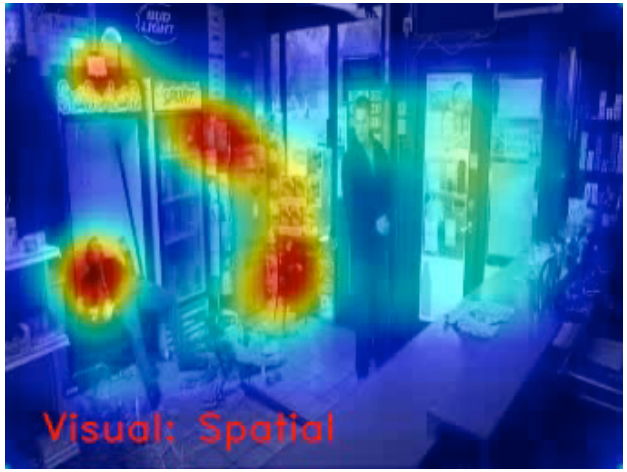
open door

walk

talk

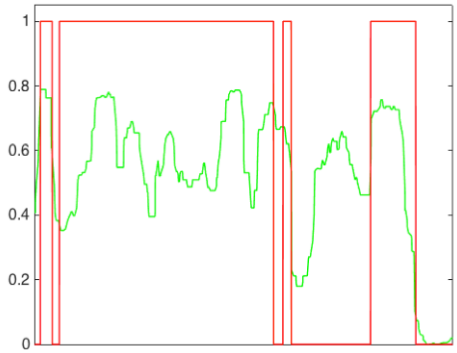
# Spatio-Temporal Visual Saliency

- **spatial saliency**
  - predict viewers fixations in image plain
  - static eye-tracking datasets: Toronto data set, MIT CAT200, SALICON, …
- **spatio-temporal saliency**
  - predict viewers fixations both in space and time
  - dynamic eye-tracking datasets: CRCNS, DIEM, DHF1K, …




# Video Summarization

- summarization task refers to producing a shorter version of a video:
  - video skims that contain only the necessary and non redundant information required for context understanding
  - human annotated importance scores per frame/segment
  - annotated video datasets from multiple people: SumMe, TVSum50, COGNIMUSE




model's imprtance score
human annotation

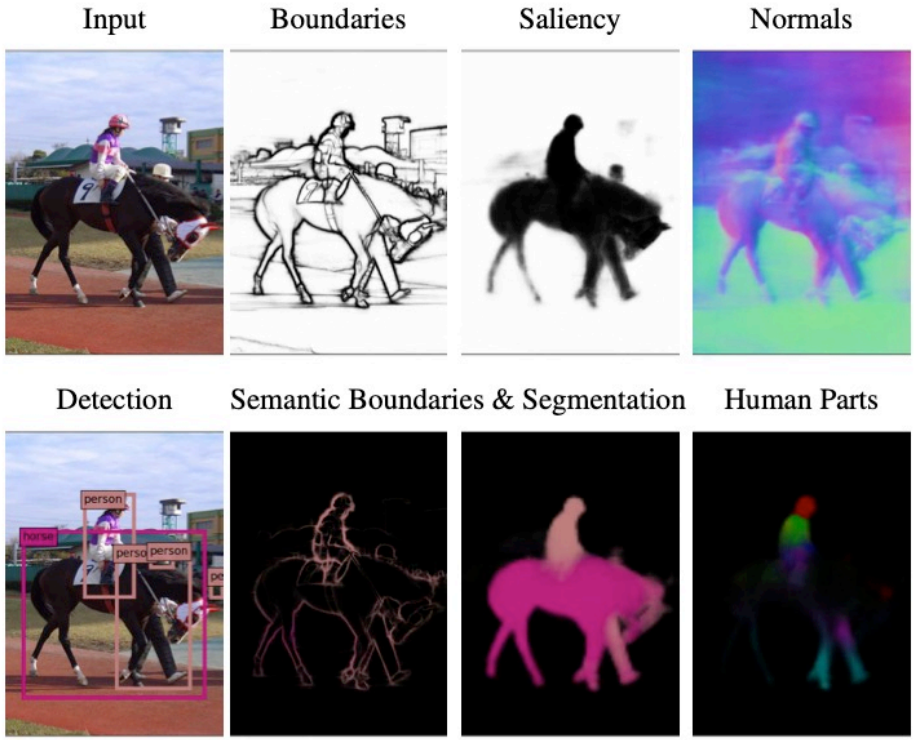
# Multi-task Networks: goal-motivation

- all these problems require the integration and modeling of the temporal evolution:
  - spatio-temporal saliency estimation ←→ *eye-tracking data*
  - visual concept understanding (i.e. actions) ←→ *annotated labels*
  - video summarization ←→ *importance scores from humans*
- can we jointly tackle these problems like humans do in the sense of:
  - **See → Understand → Summarize it**

?

# Spatial Multi-task Networks

- recent works for **spatial computer vision tasks** train multi-task networks or find the structure among visual tasks and apply transfer learning




I. Kokkinos. Ubernet: Training a universal convolutional neural network for low-, mid-, and high-level vision using diverse datasets and limited memory. In *Proc. CVPR*, 2017.

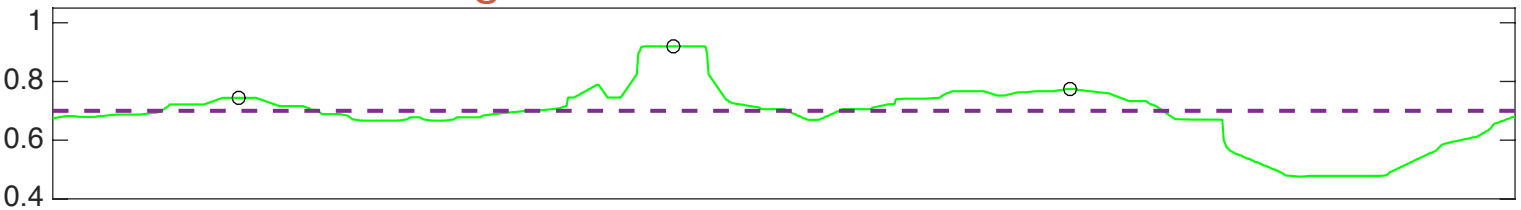
# Spatio-Temporal Multi-task Networks

- can we design and train a **single multi-task network** for solving multiple **spatio-temporal problems** with less computational resources

?

SUSiNet


Understand —> recognized action: *ride horse*




Summarize it

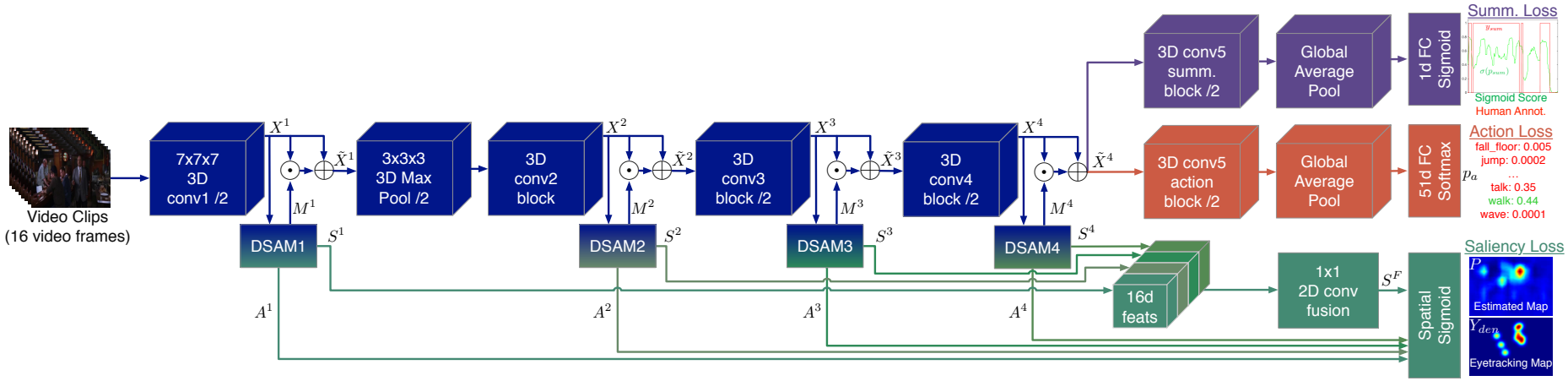
# SUSiNet Contributions

① **multi-task spatio-temporal network** that is jointly end-to-end trained for above tasks with multiple and diverse datasets:
- employs the **same video input**
- produces **multiple output types** (saliency maps or classification labels)

② **deeply supervised mechanism** through an attention module related to human attention as it is expressed by eye-tracking data

③ **extensive evaluation** on 7 different datasets:
- multi-task network performs as well as single-task methods (or in some cases better) and requires less computational budget

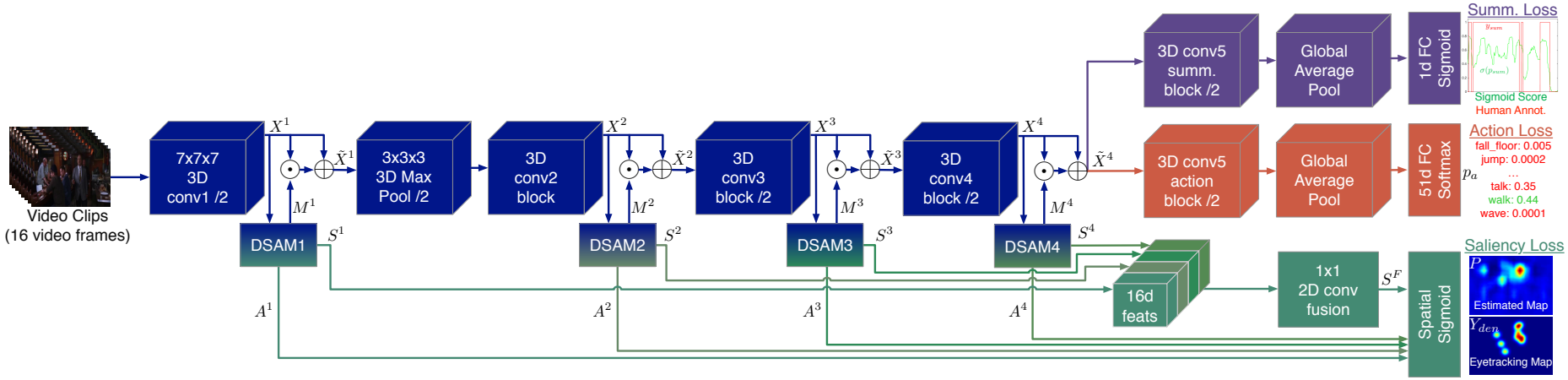
# SUSiNet architecture




$$\mathcal{L}(\mathbf{W}_{all}) = \alpha_{sal} \sum_{j \in \mathcal{B}} \mathcal{L}^j_{sal}(\mathbf{W}'_{sal}) + \alpha_{act} \sum_{j \in \mathcal{B}} \mathcal{L}^j_{act}(\mathbf{W}'_{act}) + \alpha_{sum} \sum_{j \in \mathcal{B}} \mathcal{L}^j_{sum}(\mathbf{W}'_{sum})$$

- multi-task spatio-temporal network based on 3D ResNet
  - global and task-specific modules
  - DSAM: deeply supervision attention module
- multi-task and end-to-end training

# SUSiNet architecture - saliency

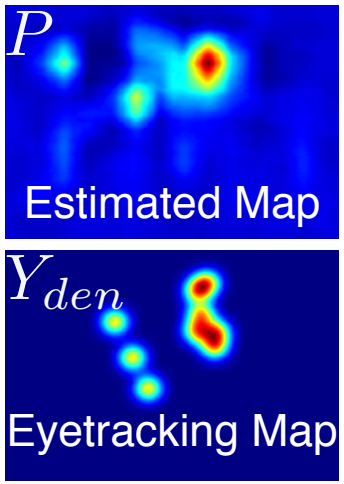



saliency

$$\mathcal{L}(\mathbf{W}_{all}) = \alpha_{sal} \sum_{j \in \mathcal{B}} \mathcal{L}^j_{sal}(\mathbf{W}'_{sal}) + \alpha_{act} \sum_{j \in \mathcal{B}} \mathcal{L}^j_{act}(\mathbf{W}'_{act}) + \alpha_{sum} \sum_{j \in \mathcal{B}} \mathcal{L}^j_{sum}(\mathbf{W}'_{sum})$$

$$\mathcal{L}^j_{sal}(\mathbf{W}'_{sal}) = w_1 \mathcal{L}^j_{CE} + w_2 \mathcal{L}^j_{CC} + w_3 \mathcal{L}^j_{NSS},$$

$$\mathcal{L}_{sal}(\mathbf{W}'_{sal}) = \mathcal{D}(\mathbf{W}'_{sal} | \sigma(S^F), Y_{sal}) + \sum_{m=1}^{4} \mathcal{D}(\mathbf{W}^m_{AM} | \sigma(A^m), Y_{sal}),$$



# SUSiNet architecture – saliency losses

$$\mathcal{L}_{sal}(\mathbf{W}'_{sal}) = \mathcal{D}(\mathbf{W}'_{sal}|\sigma(S^F), Y_{sal}) + \sum_{m=1}^{4} \mathcal{D}(\mathbf{W}^m_{AM}|\sigma(A^m), Y_{sal}),$$

$$\tilde{\mathcal{D}}_{CE}(\mathbf{W}|P, \tilde{Y}_{den}) = -\beta \cdot \sum_{x,y \in \mathcal{Y}_+} \log(P(x, y; \mathbf{W})) - (1-\beta) \cdot \sum_{x,y \in \mathcal{Y}_-} (1 - \log(P(x, y; \mathbf{W}))),$$

cross-entropy

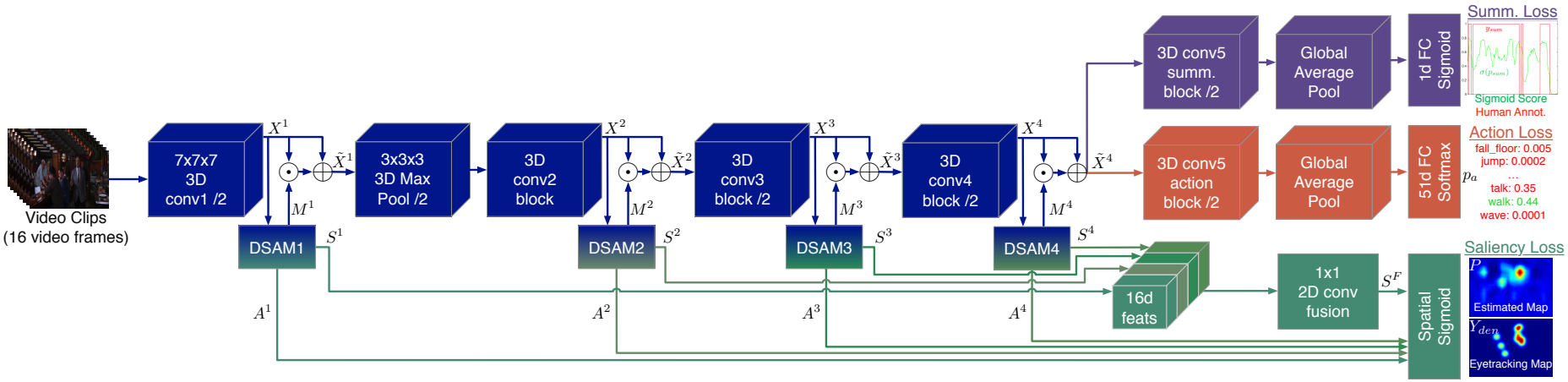
$$\mathcal{D}_{CC}(\mathbf{W}|P, Y_{den}) = -\frac{\mathrm{cov}(P(x, y; \mathbf{W}), Y_{den}(x, y))}{\rho(P(x, y; \mathbf{W})) \cdot \rho(Y_{den}(x, y))},$$

CC loss

$$\mathcal{D}_{NSS}(\mathbf{W}|\tilde{P}, Y_{fix}) = -\frac{1}{N_f} \sum_{x,y} \tilde{P}(x, y; \mathbf{W}) \odot Y_{fix}(x, y),$$

NSS loss

# SUSiNet architecture - saliency




saliency

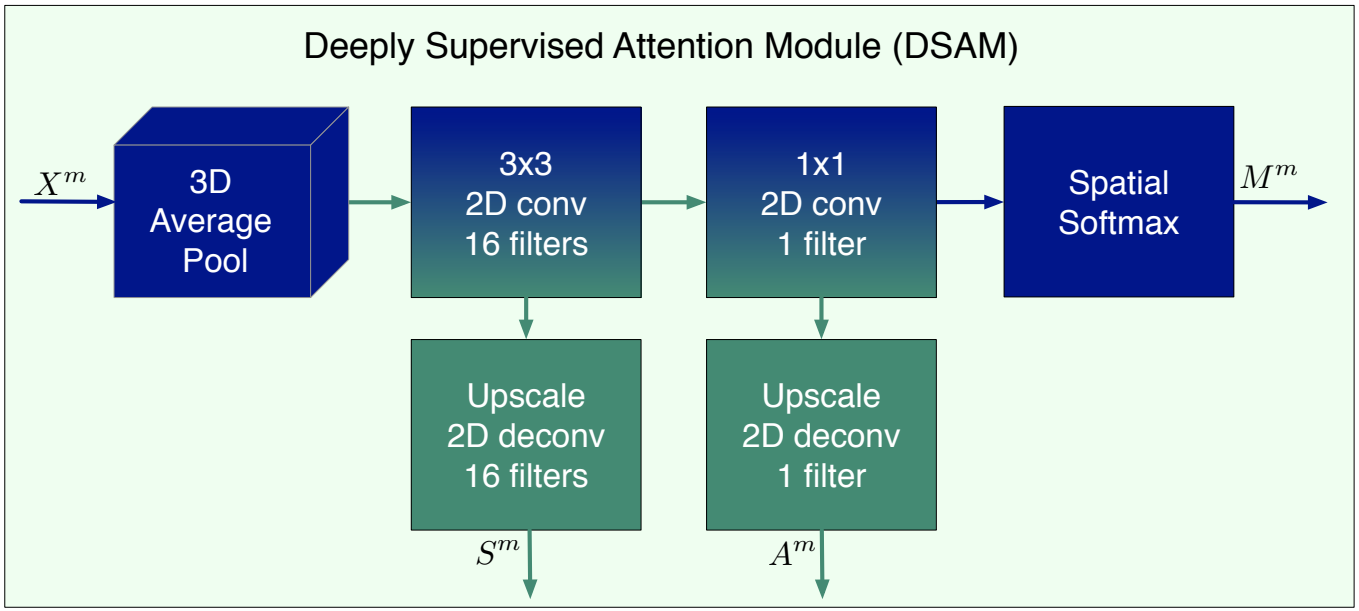
$$\mathcal{L}(\mathbf{W}_{all}) = \alpha_{sal} \sum_{j \in \mathcal{B}} \mathcal{L}^j_{sal}(\mathbf{W}'_{sal}) + \alpha_{act} \sum_{j \in \mathcal{B}} \mathcal{L}^j_{act}(\mathbf{W}'_{act}) + \alpha_{sum} \sum_{j \in \mathcal{B}} \mathcal{L}^j_{sum}(\mathbf{W}'_{sum})$$

$$\mathcal{L}^j_{sal}(\mathbf{W}'_{sal}) = w_1 \mathcal{L}^j_{CE} + w_2 \mathcal{L}^j_{CC} + w_3 \mathcal{L}^j_{NSS},$$

$$\mathcal{L}_{sal}(\mathbf{W}'_{sal}) = \mathcal{D}(\mathbf{W}'_{sal} | \sigma(S^F), Y_{sal}) + \sum_{m=1}^{4} \mathcal{D}(\mathbf{W}^m_{AM} | \sigma(A^m), Y_{sal}),$$

deep supervision

# Deeply supervised attention module




attention map

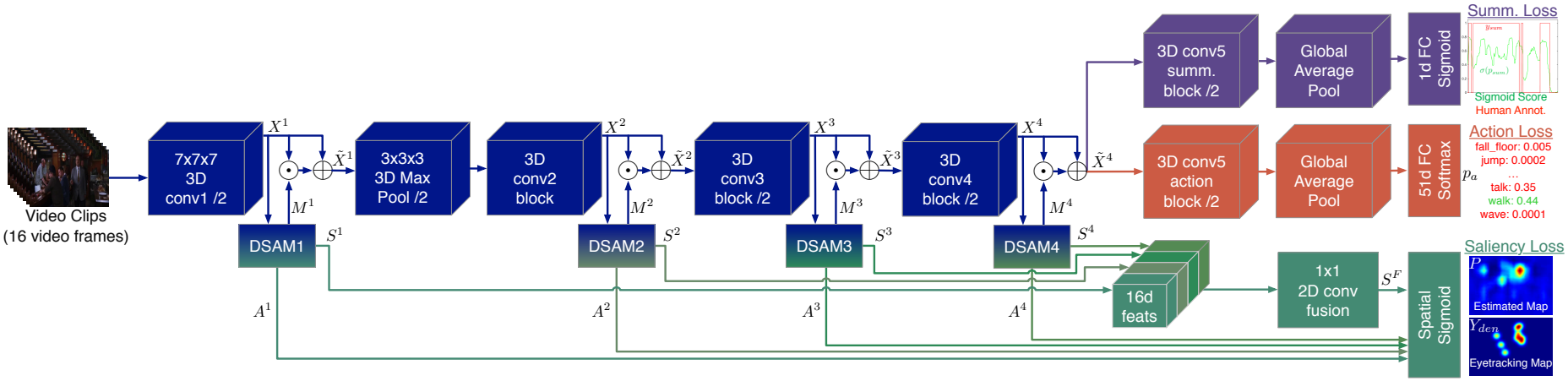
saliency features activation map

$$\tilde{X}_m = (1 + M^m) \odot X_m, \quad m = 1, \ldots, 4.$$

$$M^m(x, y) = \frac{\exp(A^m(x, y))}{\sum_x \sum_y \exp(A^m(x, y))}.$$

attention mechanism

# SUSiNet architecture - action




action

$$\mathcal{L}(\mathbf{W}_{all}) = \alpha_{sal} \sum_{j \in \mathcal{B}} \mathcal{L}^j_{sal}(\mathbf{W}'_{sal}) + \alpha_{act} \sum_{j \in \mathcal{B}} \mathcal{L}^j_{act}(\mathbf{W}'_{act}) + \alpha_{sum} \sum_{j \in \mathcal{B}} \mathcal{L}^j_{sum}(\mathbf{W}'_{sum})$$

$$\mathcal{L}^j_{act}(\mathbf{W}'_{act}) = -\log p^j_a(c_j; \mathbf{W}'_{act}),$$

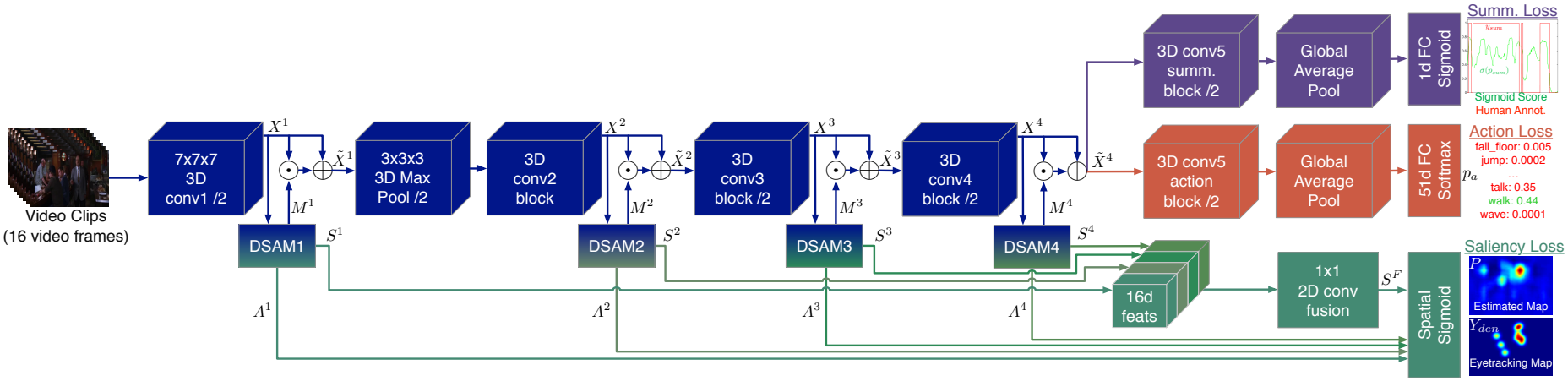
multi-class cross-entropy

fall_floor: 0.005
jump: 0.0002

$p_a$ ...

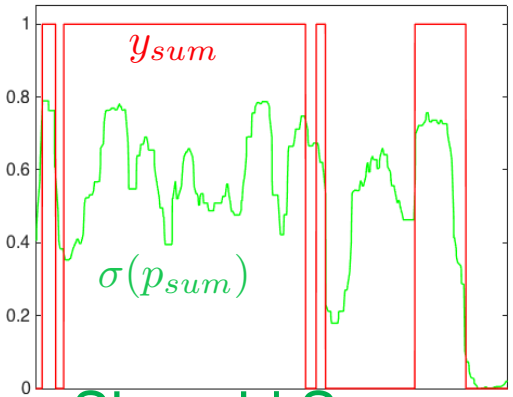
talk: 0.35
walk: 0.44
wave: 0.0001

# SUSiNet architecture - summarization




summarization

$$\mathcal{L}(\mathbf{W}_{all}) = \alpha_{sal} \sum_{j \in \mathcal{B}} \mathcal{L}^j_{sal}(\mathbf{W}'_{sal}) + \alpha_{act} \sum_{j \in \mathcal{B}} \mathcal{L}^j_{act}(\mathbf{W}'_{act}) + \alpha_{sum} \sum_{j \in \mathcal{B}} \mathcal{L}^j_{sum}(\mathbf{W}'_{sum})$$




$$\mathcal{L}^j_{sum}(\mathbf{W}'_{sum}) = -\gamma \cdot y^j_{sum} \cdot \log(\sigma(p^j_{sum}(\mathbf{W}'_{sum}))) \\ -(1 - y^j_{sum}) \cdot (1 - \log(\sigma(p^j_{sum}(\mathbf{W}'_{sum})))),$$

binary cross-entropy (BCE)

# SUSiNet: evaluation procedure

- evaluation on **7 video** datasets (3 for saliency, 2 for action, 3 for summarization)
- training using a cross-validation approach over different splits of **diverse datasets**
  - data augmentation (e.g. spatial and temporal cropping)
  - asynchronous Stochastic Gradient Descent (SGD)
- multi-task network performs **equally well** or **in some cases even better** than the single-task methods → requires **less computational budget**
  - multiple evaluation metrics

# SUSiNet evaluation results - saliency

| Method \ Dataset | | DIEM | | | | DFK1K | | | | ETMD | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | CC ↑ | NSS ↑ | AUC-J ↑ | sAUC ↑ | CC ↑ | NSS ↑ | AUC-J ↑ | sAUC ↑ | CC ↑ | NSS ↑ | AUC-J ↑ | sAUC ↑ |
| **SUSiNet (1-task)** [ST] | | **0.6138** | **2.4267** | **0.8736** | **0.6747** | **0.4676** | **2.5908** | **0.8843** | **0.6991** | **0.5523** | **2.8365** | **0.9173** | **0.7312** |
| **SUSiNet (multi)** [ST] | | **0.5614** | **2.1398** | **0.8810** | **0.6736** | **0.4116** | **2.2092** | **0.8910** | **0.6980** | **0.4780** | **2.3642** | **0.9162** | **0.7272** |
| Deep-Net | | 0.4305 | 1.6238 | 0.8401 | 0.6262 | 0.2969 | 1.5804 | 0.8421 | 0.6432 | 0.3438 | 1.6523 | 0.8712 | 0.6588 |
| DVA | | 0.5179 | 2.1607 | 0.8599 | 0.6400 | 0.3593 | 2.0644 | 0.8609 | 0.6572 | 0.4228 | 2.2507 | 0.8848 | 0.6843 |
| SAM | | 0.5352 | 2.2482 | 0.8651 | 0.6429 | 0.3684 | 2.1180 | 0.8680 | 0.6562 | 0.4345 | 2.3155 | 0.8890 | 0.6875 |
| ACLNet | [ST] | 0.5626 | 2.2168 | 0.8717 | 0.6228 | 0.4167 | 2.2962 | 0.8883 | 0.6523 | 0.4508 | 2.2058 | 0.9073 | 0.6482 |
| DeepVS | [ST] | 0.4885 | 2.0352 | 0.8448 | 0.6248 | 0.3500 | 1.9680 | 0.8561 | 0.6405 | 0.4316 | 2.3030 | 0.8955 | 0.6672 |

- evaluation on 3 eye-tracking video datasets:
  - DIEM, DFK1K, ETMD (Eye-Tracking Movie Database)
- 4 widely used evaluation metrics:
  - CC, NSS, AUC-J, sAUC
- compare performance against 5 state-of-the-art deep learning methods :
  - both spatial and spatio-temporal models

# SUSiNet evaluation results – action recognition

| Method | Aver. Accuracy |
|---|---|
| **SUSiNet (1-task)** | **60.2** |
| **SUSiNet (multi)** | **62.7** |
| C3D | 51.6 |
| 3D ResNet-18 | 56.4 |
| 3D ResNet-50 | 61.0 |
| 3D ResNeXt-101 | 63.8 |
| RGB I3D (64f) | 66.4 |

- evaluation on all splits of HMDB51
- compare performance against several other approaches based on 3D CNN networks
- multi-task SUSiNet outperforms the single-task as well as several state-of-the-art methods

# SUSiNet evaluation results – video summarization

| Method | SumMe (F-score) | TVSum50 (F-score) |
|---|---|---|
| **SUSiNet (1-task)** | **41.10** | **59.20** |
| **SUSiNet (multi)** | **40.80** | **57.00** |
| vsLSTM | 37.6 [41.6] | 54.2 [57.9] |
| HSA-RNN | 44.1 | 59.8 |
| SEQ2SEQ | 40.8 | 56.3 |
| SUM-FCN | 47.5 [51.1] | 56.8 [59.2] |

- evaluation over the SumMe and TVSum50 datasets
- evaluation protocol based on the F-score
  - keyshot-based summary
- SUSiNet performs very close to its single-task variant
- 3D network outperforms many methods based on the sequential estimation of the clip based importance score

# COGNIMUSE Database
## Saliency, Semantic & Cross-Media Events Database

*http://cognimuse.cs.ntua.gr/database*

including:

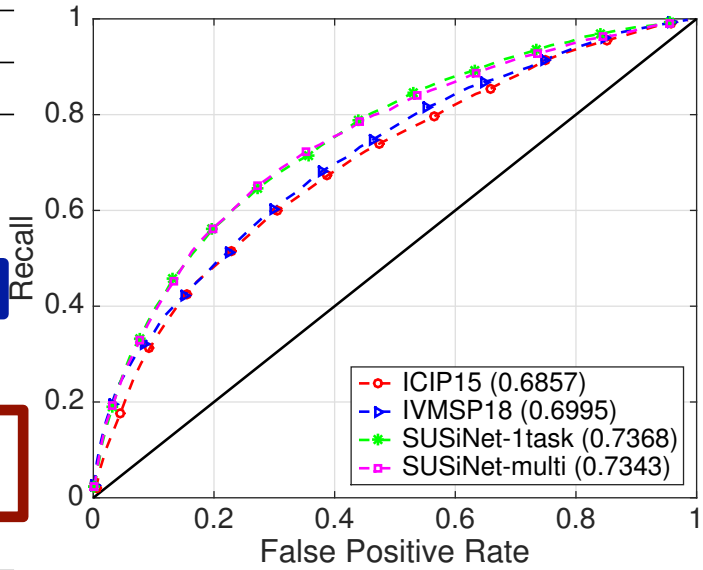
- framewise importance annotation on multiple layers
- audio & visual events annotation
- COSMOROE cross-media relations annotation
- Emotion annotation
- ETMD: eye-tracking annotations for the COGNIMUSE videos

database content:

- **7** 30-min **movie clips** from: Beautiful Mind (BMI), Chicago (CHI), Crash (CRA), The Departed (DEP), Gladiator (GLA), Lord of the Rings III: The return of the king(LOR), Finding Nemo (FNE)
- **1** 100-min **movie**: Gone with the Wind (GWTW)

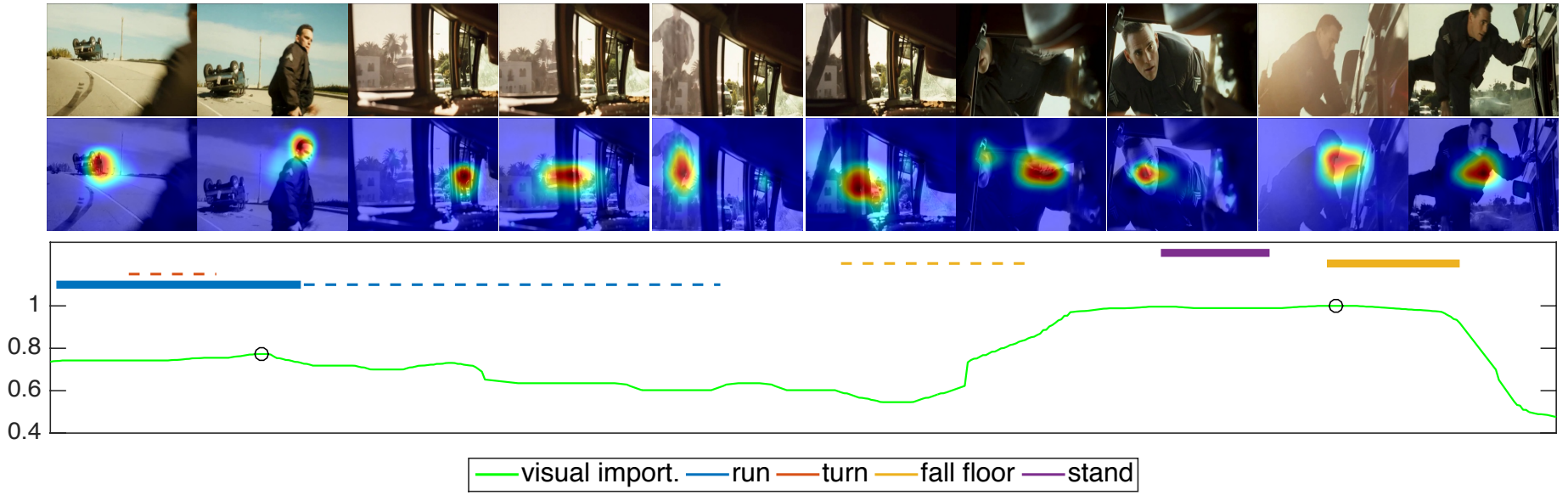
# evaluation results – COGNIMUSE database

| Task | Saliency (sAUC) | | Action (Acc.) | | Summar. (AUC) | |
|---|---|---|---|---|---|---|
| SUSiNet | 1-task | multi | 1-task | multi | 1-task | multi |
| BMI | - | - | **51.54** | 49.88 | 0.7831 | **0.8023** |
| GLA | **0.6859** | 0.6727 | **48.92** | 46.77 | **0.7863** | 0.7843 |
| CHI | **0.7601** | 0.7565 | 49.41 | **50.82** | **0.7901** | 0.7826 |
| FNE | 0.7224 | **0.7236** | - | - | **0.5490** | 0.5306 |
| LOR | 0.7297 | **0.7325** | 50.70 | **54.93** | **0.7602** | 0.7557 |
| CRA | 0.7056 | **0.7058** | **49.83** | 47.83 | **0.7424** | 0.7105 |
| DEP | **0.7837** | 0.7721 | 58.86 | **60.76** | 0.8069 | **0.8279** |
| GWW | - | - | 36.24 | **37.70** | 0.6762 | **0.6806** |
| Aver. | **0.7312** | 0.7272 | 49.36 | **49.81** | **0.7368** | 0.7343 |




- evaluation over multi-task COGNIMUSE db:
  - saliency, action, summarization
  - in many cases multi-task network **achieves better performance**
  - **low performance** for FNE → no other animation movie in training set
  - SUSiNet outperforms the two other state-of-the-art methods for the summarization task according to the ROC-AUC metric

# SUSiNet: demo video




with dotted lines the annotated actions that are not correctly recognized

# Part 1: Conclusions

- Cover **state-of-the-art approaches** for video processing and especially for action recognition
  - classic computer vision methods (i.e. dense trajectories)
  - modern CNN-based approaches
- Present **multi-task spatio-temporal** network that can jointly tackle the multiple spatio-temporal problems
  - common 3D network architecture for all tasks
  - multi-task network performs equally well or even better than the single-task methods → requires less computational budget
- Future work:
  - explore audio-visual multi-task network in order to handle the multi-modal aspects of these tasks

For more information, demos, and current results: http://cvsp.cs.ntua.gr and http://robotics.ntua.gr