Automatic Human Action Recognition using Deep Convolutional Neural Networks

Evangelos Nikoloudakis

ECE NTUA, CVSP LAB

July 2017

Action Hierarchy:

1. Action primitives
   → Individual movement of a body part
2. Actions
   → Sequence of action primitives
3. Activities
   → Sequence of actions

Action Hierarchy:

1. Action primitives
   $\rightarrow$ Individual movement of a body part

2. **Actions**
   $\rightarrow$ **Sequence of action primitives**

3. Activities
   $\rightarrow$ Sequence of actions

Action Hierarchy:

1. Action primitives
   $\rightarrow$ Individual movement of a body part
2. **Actions** $\ni$ **Gestures**
   $\rightarrow$ **Sequence of action primitives**
3. Activities
   $\rightarrow$ Sequence of actions

Action Recognition Outline

Typically:

Action Recognition = Action Localization + Action Classification

Practically:

Action Recognition = ~~Action Localization +~~ Action Classification

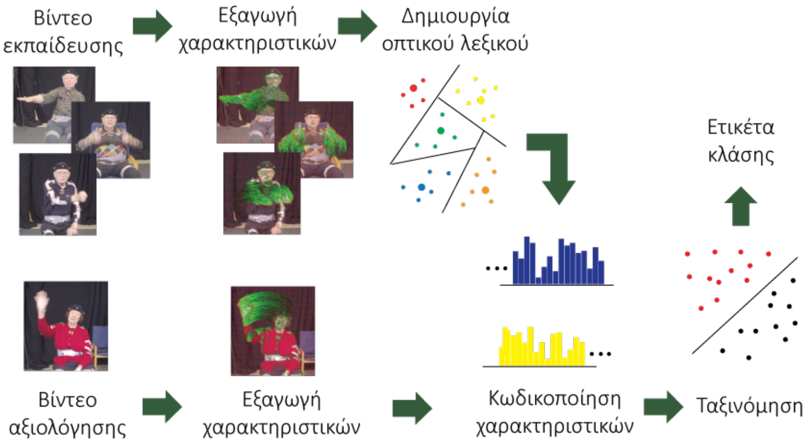
$$\Downarrow$$

Action Recognition = Action Classification

Action Recognition Challenges

- Variation in viewpoint
- Possible occlusions
- Camera motion
- Cluttered background
- Anthropometric variations
- Execution rate

Human Action Recognition System



Available action databases

- **KTH**
  - 6 classes
  - 25 persons / 4 scenarios
  - 2391 videos with 160×140 resolution
- **UCF101**
  - 101 classes
  - 13320 realistic Youtube videos
  - 3 splits: 80-110 training, 30-45 testing videos from each class
- **Hollywood2**
  - 12 classes
  - 1707 videos from 69 Hollywood movies
  - 823 training videos - 884 testing videos
- **HMDB51**
  - 51 classes
  - 6766 videos mostly from movies
  - 3 splits: 70 training, 30 testing videos from each class

COGNIMUSE database

Temporal segmentation of annotated action clips from 7 movies:

- *Beautiful Mind (BMI)* – 2001
- *Chicago (CHI)* – 2002
- *Crash (CRA)* – 2004
- *The Departed (DEP)* – 2006
- *Gladiator (GLA)* – 2000
- *Lord of the Rings (LOR)* – 2003
- *Gone with the Wind (GWW)* – 1939

Keep the 20 classes that contain at least $N_{thres} = 30$ videos. Totally 2238 videos **distributed unequally** in the 20 classes.

1. Hand-crafted techniques

2. Deep learning techniques

1. Hand-crafted techniques
   - Spatio-temporal Interest Points (STIPs)

2. Deep learning techniques

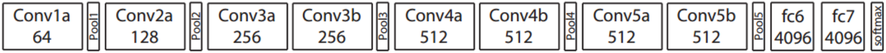
1. Hand-crafted techniques
   - Spatio-temporal Interest Points (STIPs)
   - Dense Trajectories
2. Deep learning techniques

1. Hand-crafted techniques
    - Spatio-temporal Interest Points (STIPs)
    - Dense Trajectories
2. Deep learning techniques
    - Convolutional Neural Networks

3D ConvNets

Convolutional Layers with 3D filters

⇓ **C3D architecture**

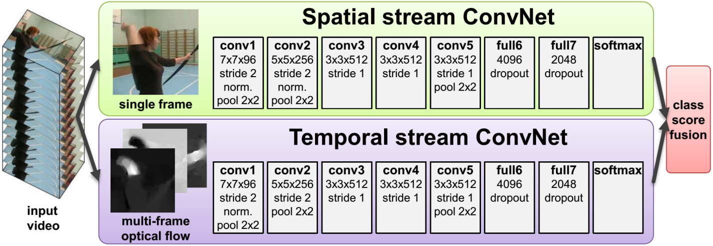


3D filters in Conv Layers
3D Pooling Layers

After training C3D net can be used as feature extractor:

1. A video is split into 16-frame long clips
2. Clips are passed to the C3D net to extract *fc6* activations
3. The clips activations are averaged to form a 4096-dim descriptor
4. This vector is then followed by L2-normalization

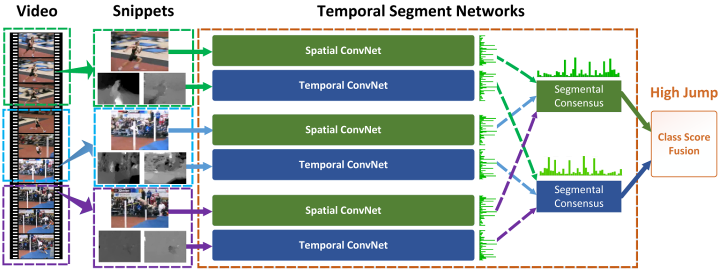
Two-Stream ConvNets



- *Spatial stream* operates on individual video frames $I_\tau \in \mathbb{R}^{w \times h \times 3}$
- *Temporal stream* operates on stacks of optical flow fields d_t^x, d_t^y :

$$I_\tau(u, v, 2k-1) = d^x_{\tau+k-1}(u, v)$$
$$I_\tau(u, v, 2k) = d^y_{\tau+k-1}(u, v), \quad u = [1; w], v = [1; h], k = [1; L]$$

Temporal Segment Networks



$$TSN\,\underbrace{(T_1, T_2, ..., T_K)}_{snippets} = H(G(\underbrace{F(T_1;\mathbf{W})}_{ConvNet}, F(T_2;\mathbf{W}), ..., F(T_K;\mathbf{W})))$$

G: segmental consensus function $\rightarrow$ average

H: prediction function $\rightarrow$ softmax classifier

Loss function: $L(y, \mathbf{G}) = -\sum_{i=1}^{C} y_i \left(G_i - \log \sum_{j=1}^{C} \exp G_j \right)$

Experiments - Results

COGNIMUSE evaluation

Feature Extraction

- improved trajectories $\rightarrow$ combined descriptor
  - $L = 15, N = 32, n_\sigma = 2, n_\tau = 3$
  - BoW encoding - codebook of 4000 K-means centroids
- C3D features
  - pre-trained model on I380K and fine-tuned on Sports-1M
  - 16-frame long non-overlapped clips
  - average of fc6 activations followed by L2 normalization

COGNIMUSE evaluation

Feature Extraction

- improved trajectories $\rightarrow$ combined descriptor
  - $L = 15, N = 32, n_\sigma = 2, n_\tau = 3$
  - BoW encoding - codebook of 4000 K-means centroids
- C3D features
  - pre-trained model on I380K and fine-tuned on Sports-1M
  - 16-frame long non-overlapped clips
  - average of fc6 activations followed by L2 normalization

Classification

- multi-class x^2 SVM
  - kernel fusion

COGNIMUSE evaluation

- 80% of samples for training (5 iterations)
  - 10 smaller classes ($\sim 30 - 60$ samples per class)

Split	iDT	C3D	C3D+iDT
10_small_classes	55.2	47.3	**58.4**

  - 10 bigger classes ($\sim 60 - 200$ samples per class)

Split	iDT	C3D	C3D+iDT
10_big_classes	51.2	44.7	**54.1**

$$\Downarrow$$

  - 8 classes (Remove *turn,walk* $\rightarrow \sim 60 - 100$ videos per class)

Split	iDT	C3D	C3D+iDT
8_classes	52.7	51.6	**59.8**

cry, pick, point something, ride horse, run, smile, stand up, wave hands

Use of 3D ConvNets

C3D net: Pre-trained on I380K and fine-tuned on Sports-1M

- Task-specific fine-tuning + end-to-end evaluation
  - Fine-tuning on UCF101, HMDB51, Hollywood2 training sets
  - Evaluation on corresponding validation sets

Database	C3D fine-tuned	C3D+iDT
UCF101	83.4	**86.7**
HMDB51	53.9	**59.6**
Hollywood2	50.7	**61.4**

Note: For UCF101, HMDB51 we calculate the average precision of the 3 splits

Use of 3D ConvNets

C3D net: Pre-trained on I380K and fine-tuned on Sports-1M

- Task-specific fine-tuning + end-to-end evaluation
  - Fine-tuning on UCF101, HMDB51, Hollywood2 training sets
  - Evaluation on corresponding validation sets

Database	C3D fine-tuned	C3D+iDT
UCF101	83.4	**86.7**
HMDB51	53.9	**59.6**
Hollywood2	50.7	**61.4**

Note: For UCF101, HMDB51 we calculate the average precision of the 3 splits

Deep-learned feature extraction is preferable!

→ Combination with other feature representations

C3D net: Pre-trained on I380K and fine-tuned on Sports-1M

- fc6 feature extraction from 16-frame long non-overlapped clips
  1. average pooling
  2. max pooling
  3. multiplication pooling

+ L2 normalization

Database	average	max	multipl.
HMDB51	52.8	**53.4**	47.2
Hollywood2	46.2	**48.1**	29.3
Congimuse	51.6	**53.8**	51.4

Use of 3D ConvNets

C3D net: Pre-trained on I380K and fine-tuned on Sports-1M

- fc6 feature extraction from 16-frame long non-overlapped clips
  1. average pooling
  2. max pooling
  3. multiplication pooling

+ L2 normalization

Database	average	max	multipl.
HMDB51	52.8	**53.4**	47.2
Hollywood2	46.2	**48.1**	29.3
Congimuse	51.6	**53.8**	51.4

- fc6 feature extraction from 8-frame overlapped clips

Database	C3D	C3D_ovrlp	C3D_ovrlp + iDT
HMDB51	53.4	**53.9**	**60.4**
Hollywood2	48.1	**48.6**	**61.9**
Cognimuse	53.8	**54.3**	**60.2**

Use of Two-stream ConvNets

Architecture: Temporal Segment Networks

- Task-specific training + end-to-end evaluation
  - Fine-tuning on UCF101, HMDB51 training sets
  - Evaluation on corresponding validation sets

Database	CNNs	Split 1	Split 2	Split 3	Average
UCF101	Spatial Stream	85.2	84.8	85.0	85.0
	Temporal Stream	87.5	90.2	90.3	89.3
	Two-Stream	93.2	94.4	93.8	**93.8**
HMDB51	Spatial Stream	53.8	49.9	48.7	50.8
	Temporal Stream	62.2	63.0	63.6	62.9
	Two-Stream	69.2	67.1	68.0	**68.1**

Use of Two-stream ConvNets

Architecture: Temporal Segment Networks

Two-stream net: Trained on HMDB51 split 1

- Spatio-temporal feature extraction from global pooling layer
  1. Sample 25 RGB frames or optical flow stacks
  2. Crop 4 corners & 1 center & their horizontal flipping
  3. Extract global pooling activations from each net
  4. Average the activations of the crops to form a 1024-dim vector
  5. Average the vectors of the 25 sampled inputs
  6. Apply L2 normalization to form a 1024-dim descriptor for the video

Use of Two-stream ConvNets

Architecture: Temporal Segment Networks

Two-stream net: Trained on HMDB51 split 1

- Spatio-temporal feature extraction from global pooling layer
  1. Sample 25 RGB frames or optical flow stacks
  2. Crop 4 corners & 1 center & their horizontal flipping
  3. Extract global pooling activations from each net
  4. Average the activations of the crops to form a 1024-dim vector
  5. Average the vectors of the 25 sampled inputs
  6. Apply L2 normalization to form a 1024-dim descriptor for the video

- x^2 multi-class SVM classification

Use of Two-stream ConvNets

Architecture: Temporal Segment Networks

Two-stream net: Trained on HMDB51 split 1

- Spatio-temporal feature extraction from global pooling layer
  1. Sample 25 RGB frames or optical flow stacks
  2. Crop 4 corners & 1 center & their horizontal flipping
  3. Extract global pooling activations from each net
  4. Average the activations of the crops to form a 1024-dim vector
  5. Average the vectors of the 25 sampled inputs
  6. Apply L2 normalization to form a 1024-dim descriptor for the video
- x^2 multi-class SVM classification

Database	TSN rgb	TSN flow	TSN rgb+flow
HMDB51(split1)	55.6	63.1	**70.2**
Hollywood2	51.8	63.7	**67.4**
Cognimuse	50.8	52.2	**60.5**

Use of Two-stream ConvNets

Architecture: Temporal Segment Networks

Two-stream net: Trained on HMDB51 split 1

- Combination of TSN features with other representations

Database	TSN rgb+flow	TSN + iDT	TSN + C3D + iDT
HMDB51(split1)	70.2	72.5	**73.8**
Hollywood2	67.4	71.7*	**72.4***
Cognimuse	60.5	61.9	**62.6**

Use of Two-stream ConvNets

Architecture: Temporal Segment Networks

Two-stream net: Trained on HMDB51 split 1

- Combination of TSN features with other representations

Database	TSN rgb+flow	TSN + iDT	TSN + C3D + iDT
HMDB51(split1)	70.2	72.5	**73.8**
Hollywood2	67.4	71.7*	**72.4***
Cognimuse	60.5	61.9	**62.6**

3 Two-stream nets: Trained on HMDB51 split 1,2,3

- max pooling of the 25 inputs activation vectors
- concatenation of the 3 nets feature vectors + L2 normalization

Method	HMDB51(split1)	Hollywood2	Cognimuse
TSN(3 nets) + C3D + iDT	**74.4**	**73.1***	**63.7**

Comparison with other methods

Method	HMDB51	Hollywood2
iDT+BoW	52.1	62.2
iDT+FV	57.2	64.3
VideoDarwin	63.7	73.7
Two-stream	59.4	-
TDDs	65.9	-
VGG+iDT	69.2	-
HRP+iDT	69.4	76.7
TSN	68.5	-
TLEs	71.1	-
EPT+iDT	-	**78.6**
SSN	73.8	-
Ours	**74.0**	73.1*

Final Proposed Method

- x^2 Multi-class SVM classification with kernel fusion of:
  1. *TSN rgb features* extracted from the 3 nets trained on HMDB51 3 splits, max-pooled and L2-normalized
  2. *TSN flow features* same as rgb
  3. *C3D features* extracted from pre-trained net on Sports1M, on 16-frame long clips with 8-frame overlap, max-pooled and L2-normalized
  4. *Combined descriptor* of BoW encoded iDT on a codebook of 4000 K-means centroids

Final Proposed Method

- x^2 Multi-class SVM classification with kernel fusion of:
  1. *TSN rgb features* extracted from the 3 nets trained on HMDB51 3 splits, max-pooled and L2-normalized
  2. *TSN flow features* same as rgb
  3. *C3D features* extracted from pre-trained net on Sports1M, on 16-frame long clips with 8-frame overlap, max-pooled and L2-normalized
  4. *Combined descriptor* of BoW encoded iDT on a codebook of 4000 K-means centroids

Practically: We propose a new combined descriptor of $\{\mathbf{TSN\ rgb}, \mathbf{TSN\ flow}, \mathbf{C3D}, \mathbf{TD}, \mathbf{HoG}, \mathbf{HoF}, \mathbf{MBHx}, \mathbf{MBHy}\}$

Future Work

- Apply other encoding methods on iDT
- Human Detection & Tracking
- Deep-learned feature extraction from shallower feature maps
- Different pooling methods to ensure temporal consistency
- Action Localization

Thanks for watching!