Chained Multi-stream Networks Exploiting Pose, Motion, and Appearance for Action Classification and Detection

Mohammadreza Zolfaghari, Gabriel L. Oliveira, Nima Sedaghat, and Thomas Brox
University of Freiburg
Freiburg im Breisgau, Germany
{zolfagha,oliveira,nima,brox}@cs.uni-freiburg.de

Abstract

General human action recognition requires understanding of various visual cues. In this paper, we propose a network architecture that computes and integrates the most important visual cues for action recognition: pose, motion, and the raw images. For the integration, we introduce a Markov chain model which adds cues successively. The resulting approach is efficient and applicable to action classification as well as to spatial and temporal action localization. The two contributions clearly improve the performance over respective baselines. The overall approach achieves state-of-the-art action classification performance on HMDB51, J-HMDB and NTU RGB+D datasets. Moreover, it yields state-of-the-art spatio-temporal action localization results on UCF101 and J-HMDB.

1. Introduction

Human action recognition is a complex task in computer vision, due to the variety of possible actions is large and there are multiple visual cues that play an important role. In contrast to object recognition, action recognition involves not only the detection of one or multiple persons, but also the awareness of other objects, potentially involved in the action, such as the pose of the person, and their motion. Actions can span various time intervals, making good use of videos and their temporal context is a prerequisite for solving the task to its full extent [42, 41].

The success of convolutional networks in recognition has also influenced action recognition. Due to the importance of multiple visual cues, as shown by Jhuang et al. [13], multi-stream architectures have been most popular. This trend was initiated by Simonyan and Zisserman [36], who proposed a simple fusion of the action class scores obtained with two separate convolutional networks, where one was trained on raw images and the other on optical flow. The relative success of this strategy shows that deep networks for action recognition cannot directly infer the relevant motion cues from the raw images, although, in principle, the network could learn to compute such cues.

In this paper, we propose a three-stream architecture\(^1\) that also includes pose, see Figure 1. Existing approaches model the temporal dynamics of human postures with hand-crafted features. We rather propose to compute the position of human body parts by a deep network. The spatio-temporal CNN can capture the temporal dynamics of pose. Additional losses on \(Y_{\text{Pose}}\) and \(Y_{OF}\) are used for training. The final output of the network \(Y_{\text{RGB}}\) is provided at the end of the chain.

Figure 1. The chained multi-stream 3D-CNN sequentially refines action class labels by analyzing motion and pose cues. Pose is represented by human body parts detected by a deep network. The spatio-temporal CNN can capture the temporal dynamics of pose. Additional losses on \(Y_{\text{Pose}}\) and \(Y_{OF}\) are used for training. The final output of the network \(Y_{\text{RGB}}\) is provided at the end of the chain.

\(^1\)https://lmb.informatik.uni-freiburg.de/projects/action_chain/
2. Related Work

Feature based approaches. Many traditional works in the field of action recognition focused on designing features to discriminate action classes [19, 44, 5, 18, 17]. These features were encoded with high order encodings, e.g., bag of words (BoW) [38] or Fisher vector based encodings [33], to produce a global representation for video and to train a classifier on the action labels. Recent research showed that most of these approaches are not only computationally expensive, but they also fail on capturing context and high-level information.

CNN based approaches. Deep learning has enabled the replacement of hand-crafted features by learned features, and the learning of whole tasks end-to-end. Several works employed deep architectures for classification [26, 41, 45, 34]. Thanks to their hierarchical feature representation, deep networks learn to capture localized features as well as context cues and can exploit high-level information from large scale video datasets. Baccouche et al. [2] firstly used a 3D CNN to learn spatio-temporal features from video and in the next step they employed an LSTM to classify video sequences. More recently, several CNN based works presented efficient deep models for action recognition [7, 31, 41]. Tran et al. [41] employed a 3D architecture to learn spatio-temporal features from videos.

Fusion of multiple modalities. Zisserman et al. [36] proposed a two-stream CNN to capture the complementary information from appearance and motion, each modality in an independent stream. Feichtenhofer et al. [9] investigated the optimal position within a convolution network in detail to combine the separate streams. Park et al. [30] proposed a gated fusion approach. In a similar spirit, Wang et al. [50] presented an adaptive fusion approach, which uses two regularization terms to learn fusion weights. In addition to optical flow, some works made use of other modalities like audio [50], warped flow [47], and object information [12] to capture complementary information for video classification. In the present work, we introduce a new, flexible fusion technique for early or late fusion via a Markov chain and show that it outperforms previous fusion methods.

Pose feature based methods. Temporal dynamics of body parts over time provides strong information on the performing action. Thus, this information has been employed for action recognition and localization in several works [4, 21, 43, 39]. Cheron et al. [3] used pose information to extract high-level features from appearance and optical flow. They showed that using pose information for video classification is highly effective. Wang et al. [43] used data mining techniques to obtain a representation for each video and finally, by using a bag-of-words model to classify videos. In the present work, we compute the human body layout efficiently with a deep network and learn the relevant spatio-temporal pose features within one of the streams of our action classification network.

3. Inputs to the Network

We rely on three input cues: the raw RGB images, optical flow, and human pose in the form of human body part segmentation. All inputs are provided as spatio-temporal inputs covering multiple frames.

3.1. Optical Flow

We compute the optical flow with the method from Zach et al. [52], which is a reliable variational method that runs sufficiently fast. We convert the x-component and y-component of the optical flow to a 3 channel RGB image by stacking components and magnitude of them [31]. The flow and magnitude values in the image are multiplied by 16 and quantized into the [0,255] interval [20, 31, 46, 47].

3.2. Body Part Segmentation

Encoder-decoder architectures with an up-convolutional part have been used successfully for semantic segmentation tasks [25, 24, 32, 3, 29], depth estimation [22] and optical
flow estimation [8]. For this work, we make use of Fast-Net [29], a network for human body part segmentation, which will provide our action recognition network with body pose information. Figure 2 illustrates the architecture of Fast-Net. The encoder part of the network is initialized with the VGG network [37]. Skip connections from the encoder to the decoder part ensure the reconstruction of details in the output up to the original input resolution.

We trained the Fast-Net architecture on the J-HMDB [13] and the MPII [1] action recognition datasets. J-HMDB provides body part segmentation masks and joint locations, while MPII provides only joint locations. To make body part masks compatible across datasets, we apply the following methodology, which only requires annotation for the joint locations. First, we derive a polygon for the torso from the joint locations around that area. Secondly, we approximate the other parts by ellipses scaled consistently based on the torso area and the distance between the respective joints; see second column of Fig. 3. We convert the body part segmentation into a 3 channel RGB image, mapping each label to a correspondent pre-defined RGB value.

To the best of our knowledge, we are the first who trained a convolutional network on body part segmentation for the purpose of action recognition. Figure 3 shows exemplary results of the body part segmentation technique on J-HMDB and MPII datasets. Clearly, the network provides good accuracy on part segmentation and is capable of handling images with multiple instances. The pose estimation network has a resolution of 150×150 and runs at 33 fps.

4. Action Recognition Network

4.1. Multi-stream Fusion with a Markov Chain

To integrate information from the different inputs we rely on the model of a multi-stream architecture [36], i.e., each input cue is fed to a separate convolutional network stream that is trained on action classification. The innovation in our approach is the way we combine these streams. In contrast to the previous works, we combine features from the different streams sequentially. Starting with the human body part stream, we refine the evidence for an action class with the optical flow stream, and finally apply a refinement by the RGB stream.

We use the assumption that the class predictions are conditionally independent due to the different input modalities. Consequently, the joint probability over all input streams factorizes into the conditional probabilities over the separate input streams.

In a Markov chain, given a sequence of inputs $X = \{X_1, X_2, ..., X_S\}$, we wish to predict the output sequence $Y = \{Y_1, Y_2, ..., Y_S\}$ such that $P(Y|X)$ is maximized. Due to the Markov property, $P(Y|X)$ can be decomposed:
To encapsulate the information in the initial prediction (see Figure 4-right), we use the convolutional part and first fully connected layer of the network presented in Figure 5 to encapsulate the information in the input modality, and \( \text{Net}_s \) is the fully connected part in Figure 5.

At each fusion stage, we concatenate the output of the function \( 3\text{DCNN}(\cdot) \) with the hidden state and the outputs from the previous stream and apply the non-linearity \( f \) before feeding them to \( \text{Net}_s \). Finally, at the output part, we use \( \text{Net}_s \) to predict action labels from \( h_s \). With the \( \text{softmax}(\cdot) \) function we convert these scores into (pseudo-)probabilities.

Using the above notation, we consider input modalities as \( X = \{ X_{\text{pose}}, X_{\text{OF}}, X_{\text{RGB}} \} \), and \( X_s = \{ x_i \}_{i=1}^{T} \), where \( x_i \) is the \( t \)-th frame in \( X_s \), and \( T \) is the total number of frames in \( X_s \). At the stage \( s = 1 \), by considering \( X_1 = X_{\text{pose}} \) we start with an initial hidden state and obtain an initial prediction (see Figure 4-right):

\[
\begin{align*}
  h_1 & = 3\text{DCNN}(X_{\text{pose}}) \\
  P(Y_1 | X) & = \text{softmax}(\text{Net}_1(h_1))
\end{align*}
\]

At each subsequent stage \( s \geq 2 \), we obtain a refined prediction \( y_s \) by combining the hidden state and the predictions from the previous stage.

\[
\begin{align*}
  P(Y_s | X, Y_{<s}) & = \text{softmax}(\text{Net}_s(h_s)) \\
  h_s & = f(h_{s-1}, 3\text{DCNN}(X_s), (Y_1, \ldots, Y_{s-1}))
\end{align*}
\]

where \( f \) is a non-linearity unit (ReLU), \( h_{s-1} \) denotes the hidden state from the previous stream, and \( y_s \) is the prediction of stream \( s \). For the \( 3\text{DCNN}(\cdot) \), we use the convolutional part and first fully connected layer of the network presented in Figure 5 to encapsulate the information in the input modality, and \( \text{Net}_s \) is the fully connected part in Figure 5.

The ordering of the sequence plays a role for the final performance. We compared different ordering options in our experiments and report them in the following section. The ordering that starts with the pose as input and ends with the RGB image yielded the best results.

It is worth noting that the concept of sequential fusion could be applied to any layer of the network. Here we placed the fusion after the first fully-connected layer, but the fusion could also be applied to the earlier convolutional layers.

### 4.2. Network Configuration

In all streams, we use the C3D architecture [41] as the base architecture, which has 17.5\( M \) parameters. The network has 8 three-dimensional convolution layers with kernel size of \( 3 \times 3 \times 3 \) and stride 1, 5 three-dimensional pooling layers with kernel size of \( 2 \times 2 \times 2 \) and stride 2 and two fully connected layers followed by a softmax; see Figure 5. Each stream is connected with the next stream via layer FC6; see Figure 4-right. Each stream takes 16 frames as input.

### 4.3. Training

The network weights are learned using mini-batch stochastic gradient descent (SGD) with a momentum of
0.9 and weight decay of $5e^{-4}$. We jointly optimize the whole network without truncating gradients and update the weights of each stream based on the full gradient including the contribution from the following stream. We initialize the learning rate with $1e^{-4}$ and decrease it by a factor of 10 every $2k$ for J-HMDB, $20k$ for UCF101 and NTU, and at multiple steps for HMDB51. The maximum number of iterations was $20k$ for J-HMDB, $40k$ for HMDB51 and $60k$ for the UCF101 and NTU datasets. We initialize the weights of all streams with an RGB network pre-trained on the large-scale Sports-1M dataset [15] except flow streams of J-HMDB and HMDB51 datasets which we initialize with the weights of UCF101 flow stream finetuned on Sports-1M model.

We split each video into clips of 16 frames with an overlap of 8 frames and feed each clip individually into the network stream with size of $16 \times 112 \times 112$. We apply corner cropping as a form of data augmentation to the training data. Corner cropping extracts regions from the corners and the center of the image. It helps to prevent the network from bias towards the center area of the input. Finally, we resize these cropped regions to the size of $112 \times 112$. In each iteration, all streams take the same clip from the video with the same augmentation but with different modalities as input.

We used Caffe [14] and an NVIDIA Titan X GPU to run our experiments. The training time for the J-HMDB dataset was $\sim 10$ hours for the full network.

### 4.4. Temporal Processing of the Whole Video

At test time, we feed the architecture with a temporal window of 16 frames. The stride over the video is 8. Each set of inputs is randomly selected for cropping operations, which are 4 corners and 1 center crop for the original image and their horizontal flipping counterpart. We extract scores before the softmax normalization in the last stream ($Y_{RGB}$).

In case of action classification, the final score of a video is calculated by taking the average of scores over all temporal windows across a video and 10 crop scores per clip. Apart from averaging, we also tested a multi-resolution approach, which we call multi-granular (MG), where we trained separate networks for three different temporal resolutions. These are assembled as (1) 16 consecutive frames, (2) 16 frames from a temporal window of 32 frames by a sample rate of 2, and (3) 16 frames sampled randomly from the entire video. For the final score, we take the average over the scores produced by these temporal resolution networks. This approach extends the temporal context that the network can see, which can be useful for more complex actions with longer duration.

In case of temporal action detection, we localize the action in time by thresholding the score provided for each frame. Clearly, the MG approach is not applicable here. In addition to the action score, also the human body part network helps in temporal localization: we do not detect an action as long as no human is detected. More details on the spatio-temporal action detection are provided in the experimental section and in the supplemental material.

### 5. Experiments

#### 5.1. Datasets

**UCF-101** [40] contains more than 2 million frames in more than 13,000 videos, which are divided into 101 human action classes. The dataset is split into three folds and each split contains about 8000 videos for training. The UCF101 dataset also comes with a subset for spatio-temporal action detection.

**HMDB51** [16] contains 6766 videos divided into 51 action classes, each with at least 101 samples. The evaluation follows the same protocol used for UCF-101.

**J-HMDB** contains a subset of videos from the Hmdb dataset, for which it provides additional annotation, in particular optical flow and joint localization [13]. Thus, it is well-suited for evaluating the contribution of optical flow, body part segmentation, and the fusion of all cues via a Markov chain. The dataset comprises 21 human actions. The complete dataset has 928 clips and 31838 frames. There are 3 folds for training and testing for this dataset. The videos in J-HMDB are trimmed and come with bounding boxes. Thus, it can be used also as a benchmark for spatial action localization.

---

**Figure 5.** Base architecture used in each stream of the action recognition network. We define the convolutional part and the first fully connected layer as 3DCNN and the remaining fully connected layers as $Net_s$. 

---

---
Table 1. The value of different cues and their integration for action recognition on the UCF101, HMDB51, and J-HMDB datasets (split 1). Adding optical flow and pose is always beneficial. Integration via the proposed Markov chain clearly outperforms the baseline fusion approach. In all cases, the accuracy achieved with estimated optical flow and body parts almost reaches the upper bound performance when providing ground truth values for those inputs.

<table>
<thead>
<tr>
<th>Streams</th>
<th>Variant</th>
<th>UCF101</th>
<th>HMDB51</th>
<th>J-HMDB</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>RGB</td>
<td>84.2%</td>
<td>53.3%</td>
<td>60.8%</td>
</tr>
<tr>
<td></td>
<td>OF</td>
<td>79.6%</td>
<td>45.2%</td>
<td>61.9%</td>
</tr>
<tr>
<td></td>
<td>Pose</td>
<td>56.9%</td>
<td>36.0%</td>
<td>45.5%</td>
</tr>
<tr>
<td></td>
<td>Pose (GT)</td>
<td>-</td>
<td>-</td>
<td>56.8%</td>
</tr>
<tr>
<td>RGB+OF</td>
<td>baseline</td>
<td>87.1%</td>
<td>55.6%</td>
<td>62.7%</td>
</tr>
<tr>
<td></td>
<td>chained</td>
<td>88.9%</td>
<td>61.7%</td>
<td>72.8%</td>
</tr>
<tr>
<td></td>
<td>chained+MG</td>
<td>-</td>
<td>66.0%</td>
<td>-</td>
</tr>
<tr>
<td>3 w/o GT</td>
<td>baseline</td>
<td>89.1%</td>
<td>57.5%</td>
<td>70.2%</td>
</tr>
<tr>
<td></td>
<td>chained</td>
<td>90.4%</td>
<td>62.1%</td>
<td>79.1%</td>
</tr>
<tr>
<td></td>
<td>chained+MG</td>
<td>91.3%</td>
<td>71.1%</td>
<td>-</td>
</tr>
<tr>
<td>3 with GT</td>
<td>baseline</td>
<td>-</td>
<td>-</td>
<td>72.0%</td>
</tr>
<tr>
<td></td>
<td>chained</td>
<td>-</td>
<td>-</td>
<td>83.2%</td>
</tr>
</tbody>
</table>

Table 2. Comparison to the state of the art on UCF101, HMDB51, and J-HMDB datasets (over all three splits).

<table>
<thead>
<tr>
<th>Methods</th>
<th>UCF101</th>
<th>HMDB51</th>
<th>J-HMDB</th>
</tr>
</thead>
<tbody>
<tr>
<td>TS Fusion [9]</td>
<td>92.5%</td>
<td>65.4%</td>
<td>-</td>
</tr>
<tr>
<td>LTC [42]</td>
<td>91.7%</td>
<td>64.8%</td>
<td>-</td>
</tr>
<tr>
<td>Two-stream [36]</td>
<td>88.0%</td>
<td>59.4%</td>
<td>-</td>
</tr>
<tr>
<td>TSN [47]</td>
<td>94.2%</td>
<td>69.4%</td>
<td>-</td>
</tr>
<tr>
<td>CPD [28]</td>
<td>92.3%</td>
<td>66.2%</td>
<td>-</td>
</tr>
<tr>
<td>Sym. FV+SVM [6]</td>
<td>90.6%</td>
<td>67.8%</td>
<td>-</td>
</tr>
<tr>
<td>Multi-Granular [20]</td>
<td>90.8%</td>
<td>63.6%</td>
<td>-</td>
</tr>
<tr>
<td>M-fusion [30]</td>
<td>89.1%</td>
<td>54.9%</td>
<td>-</td>
</tr>
<tr>
<td>KVMF [53]</td>
<td>93.1%</td>
<td>63.3%</td>
<td>-</td>
</tr>
<tr>
<td>P-CNN [4]</td>
<td>-</td>
<td>-</td>
<td>61.1%</td>
</tr>
<tr>
<td>Action tubes [10]</td>
<td>-</td>
<td>-</td>
<td>62.5%</td>
</tr>
<tr>
<td>TS R-CNN [31]</td>
<td>-</td>
<td>-</td>
<td>70.5%</td>
</tr>
<tr>
<td>MR-TS R-CNN [31]</td>
<td>-</td>
<td>-</td>
<td>71.1%</td>
</tr>
<tr>
<td>Ours (chained)</td>
<td>91.1%</td>
<td>69.7%</td>
<td>76.1%</td>
</tr>
</tbody>
</table>

Table 3. Comparison to literature on the NTU RGB+D benchmark.

<table>
<thead>
<tr>
<th>Method</th>
<th>Cross Subject %</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deep LSTM [35]</td>
<td>60.7%</td>
</tr>
<tr>
<td>P-LSTM [35]</td>
<td>62.93%</td>
</tr>
<tr>
<td>HOG² [27]</td>
<td>32.2%</td>
</tr>
<tr>
<td>FTP DS [11]</td>
<td>60.23%</td>
</tr>
<tr>
<td>ST-LSTM [23]</td>
<td>69.2%</td>
</tr>
<tr>
<td>Ours (Pose)</td>
<td>67.8%</td>
</tr>
<tr>
<td>Ours (RGB+OF+Pose - Baseline)</td>
<td>76.9%</td>
</tr>
<tr>
<td>Ours (RGB+OF+Pose - Chained)</td>
<td>80.8%</td>
</tr>
</tbody>
</table>

NTU RGB+D is a recent action recognition dataset that is quite large and provides depth and pose ground truth [35]. It contains more than 56,000 sequences and 4 million frames. NTU provides 60 action classes and 3D coordinates for 25 joints. Additionally, the high intra-class variations make NTU one of the most challenging datasets.

5.2. Action Classification

Table 1 shows that fusion with the sequential Markov chain model outperforms the baseline fusion consistently across all datasets. The baseline fusion is shown in Figure 4 and can be considered a strong baseline. It consists of fusing the multiple modalities through feature concatenation followed by a set of fully connected layers. The network is trained jointly.

Adding pose leads to a substantial improvement over the two-stream version. This confirms that pose plays an important role as complementary modality for action recognition tasks. Again, the Markov chain fusion is advantageous with a large margin.

For the J-HMDB dataset, ground truth for optical flow and pose is available and can be provided to the method. While not being relevant in practice, running the recognition with this ground truth shows on how much performance is lost due to erroneous optical flow and pose estimates. Surprisingly, the difference between the results is rather small, showing that the network does not suffer much from imperfect estimates. This conclusion can be drawn independently of the fusion method.

Finally, the temporal multi-granularity fusion (MG) further improves results. Especially on HMDB51, there is a large benefit.

5.2.1 Comparison with the state-of-the-art

Table 2 compares the proposed network to the state of the art in action classification. In contrast to Table 1, the comparison does not show the direct influence of single contributions anymore, since this table compares whole systems that are based on quite different components. Many of these systems also use other features extraction approaches, such as improved dense trajectories (IDT), which generally have a positive influence on the results, but also make the system more complicated and harder to control. Our network outperforms the state of the art on J-HMDB, NTU, and HMDB51. Also, on UCF101 dataset our approach is on par with the current state of the art while it does not rely on any additional hand-crafted features. In two stream case (RGB+OF), if we replace the 3DCNN network by the TSN approach [47], we obtain a classification accuracy of 94.05% on UCF101 (over 3 splits), which is the state of the art also on this dataset. However, the TSN approach does not allow for action detection anymore.

Finally, we ran the network on the recent NTU RGB+D dataset, which is larger and more challenging than the previous datasets. The dataset is popular for the evaluation of
methods that are based on human body pose. Clearly, the result of our network, shown in Table 3, compares favorably to the existing methods. As a result, the used pose estimation network is competitive with pose estimates using depth images and that our way to integrate this information with the raw images and optical flow is advantageous.

5.2.2 **Ordering of modalities in the Markov chain.**

Table 4 shows an analysis on how the order of the modalities affects the final classification accuracy. Clearly, the ordering has an effect. The proposed ordering starting with the pose and then adding the optical flow and the RGB images performed best, but there are alternative orders that do not perform much worse.

Table 5 quantifies the improvement in accuracy when adding a modality. Clearly, each additional modality improves the results.

5.2.3 **Fusion location**

In principle the chained fusion can be applied to any layer in the network. We studied the effect of this choice. In contrast to the large scale evaluation in Feichtenhofer et al. [9], we tested only two locations: FC6 and FC7. Table 6 shows a clear difference only on the J-HMDB dataset. There it seems that an earlier fusion, at a level where the features are not too abstract yet, is advantageous. This is similar to the outcome of the study by Feichtenhofer et al. [9], where the last convolutional layer worked best.

5.2.4 **Effect of clip length**

We analyzed the effect of the size of the temporal window on the action recognition performance. Larger windows clearly improve the accuracy on all datasets; see Table 7. For the J-HMDB dataset (RGB modality) we use a temporal window ranging from 4 to 16 frames every 4 frames. The highest accuracy is obtained with a 16 frames clip size. Based on the J-HMDB minimum video size, 16 is the highest possible time frame to be explored. We also tested mul-
Figure 6. Scheme for spatio-temporal action detection. The chained network provides action class scores and body part segmentations per frame. From these we compute action tubes and their actionness scores; see the supplemental material for details.

Table 8. Spatial action detection results (Video mAP) on the J-HMDB dataset. Across all IoU thresholds, our model outperforms the state of the art.

<table>
<thead>
<tr>
<th>IoU threshold (δ)</th>
<th>0.1</th>
<th>0.2</th>
<th>0.3</th>
<th>0.4</th>
<th>0.5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actionness [46]</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>56.4</td>
</tr>
<tr>
<td>ActionTubes [10]</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>53.3</td>
</tr>
<tr>
<td>Weinzaepfel et al. [48]</td>
<td>63.1</td>
<td>63.5</td>
<td>62.2</td>
<td>60.7</td>
<td></td>
</tr>
<tr>
<td>Peng et al. [31]</td>
<td>74.3</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>73.1</td>
</tr>
<tr>
<td>Ours</td>
<td>78.8</td>
<td>78.2</td>
<td>77.1</td>
<td>75.0</td>
<td>73.4</td>
</tr>
</tbody>
</table>

Table 9. Spatio-temporal action detection results (Video mAP) on UCF101 dataset (split1). Across all IoU thresholds, our model outperforms the state of the art.

<table>
<thead>
<tr>
<th>IoU threshold (δ)</th>
<th>0.05</th>
<th>0.1</th>
<th>0.2</th>
<th>0.3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Weinzaepfel et al. [48]</td>
<td>54.28</td>
<td>51.68</td>
<td>46.77</td>
<td>37.82</td>
</tr>
<tr>
<td>Yu et al. [51]</td>
<td>42.80</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Peng et al. [31]</td>
<td>54.46</td>
<td>50.39</td>
<td>42.27</td>
<td>32.70</td>
</tr>
<tr>
<td>Weinzaepfel et al. [49]</td>
<td>62.8</td>
<td>-</td>
<td>45.4</td>
<td>-</td>
</tr>
<tr>
<td>Ours</td>
<td>65.22</td>
<td>59.52</td>
<td>47.61</td>
<td>38.00</td>
</tr>
</tbody>
</table>

6. Conclusions

We have proposed a network architecture that integrates multiple cues sequentially via a Markov chain model. We have shown that this sequential fusion clearly outperforms other ways of fusion, because it can consider the mutual dependencies of cues during training while avoiding over-fitting due to very large network models. Our approach provides state-of-the-art performance on all four challenging action classification datasets UCF101, HMDB51, J-HMDB and NTU RGB+D while not using any additional hand-crafted features. Moreover, we have demonstrated the value of a reliable pose representation estimated via a fast convolutional network. Finally, we have shown that the approach generalizes also to spatial and spatio-temporal action detection, where we obtained state-of-the-art results as well.

7. Acknowledgements

We acknowledge funding by the ERC Starting Grant VideoLearn and the Freiburg Graduate School of Robotics.
References


[38] K. Soomro, H. Idrees, and M. Shah. Predicting the where and what of actors and actions through online action localization. In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2016. 2


