Dynamic Motion Representation for Human Action Recognition*

Sadjad Asghari-Esfeden
Smartvid.io, Northeastern University
sadjad@ece.neu.edu

Mario Sznaier
Northeastern University
sznaier@coe.neu.edu

Octavia Camps
Northeastern University
camps@coe.neu.edu

Abstract

Despite the advances in Human Activity Recognition, the ability to exploit the dynamics of human body motion in videos has yet to be achieved. In numerous recent works, researchers have used appearance and motion as independent inputs to infer the action that is taking place in a specific video. In this paper, we highlight that while using a novel representation of human body motion, we can benefit from appearance and motion simultaneously. As a result, better performance of action recognition can be achieved. We start with a pose estimator to extract the location and heat-map of body joints in each frame. We use a dynamic encoder to generate a fixed size representation from these body joint heat-maps. Our experimental results show that training a convolutional neural network with the dynamic motion representation outperforms state-of-the-art action recognition models. By modeling distinguishable activities as distinct dynamical systems and with the help of two stream networks, we obtain the best performance on HMDB, JHMDB, UCF-101, and AVA datasets.

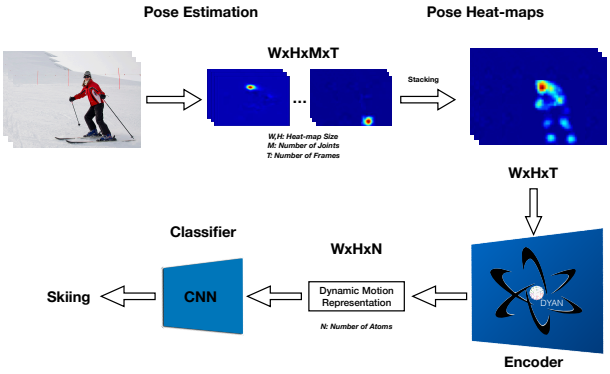

Figure 1. Human body pose encapsulates useful information to recognize the human action. Given a video, we extract pose heat-map sequences and encode them with a dynamic based model to achieve a comprehensive video representation for action classification.

1. Introduction

In recent years, the computer vision community has made significant progress in the field of action recognition and localization, thanks to large real-world human action datasets. In addition to many advancements, datasets such as UCF101 [32], HMBD51 [52], Kinetics [30], Moments in Time [40], Something Something [16], Charades [47, 48], HACS [72], DALY [62], YouTube-8M [1], Human 3.6M [8, 24], Hollywood [39], NTU [45], UCF-Sports [51, 42], and AVA [17] have led this task to a more challenging and realistic problem. Also, challenges like THUMOS [23, 15] and ActivityNet [13] have significantly contributed to the advancement of different tasks in video analysis.

*This work was supported in part by NSF grants IIS–1814631, ECCS–1808381, and CMMI–1638234; and the Alert DHS Center of Excellence under Award Number 2013-ST-061-ED0001. The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the U.S. Department of Homeland Security.

The early deep learning based approaches addressing video classification primarily employed end-to-end sequential Convolutional Neural Networks (CNNs) in concatenation with Recurrent Neural Networks (RNNs), in order to first capture appearance-based representations for action prediction [11].

RNNs and Long-Short-Term Memory Networks (LSTMs) have appeared to perform well in text-related tasks such as speech recognition, language modeling, translation, and image captioning [20, 27]. However, their use for action recognition has yet to show significant improvements. CNNs are very successful dealing with still images in tasks such as image classification [31] and object detection and segmentation [36], but there is a lot of room to work on a sequence of images (frames) when processing video clips.

Newer approaches have demonstrated the importance of incorporating motion information to CNNs, by introducing a two-stream architecture [49, 60] that trains networks in

978-1-7281-6553-0/20/$31.00 ©2020 IEEE

parallel for separate streams of still RBG images and stacks of optical flow. Two-stream architectures are indeed beneficial for video action recognition, since some activities could be captured uniquely based on the appearance of the still images and their context (for example swimming and basketball), whereas others might have divergent presentations but similar dynamic cues (as in speaking and listening).

Nevertheless, very deep convolutional networks do not exploit the dynamical structure present in both appearance-based and motion-based feature maps by assuming a priori that these representations will fall into very deep model distributions.

Adding information other than raw RGB frames can be a great help to the task of activity recognition, as some of the datasets [17] include audio as an additional modality to the video sequences. Another example using different modalities in action datasets is the addition of 2D/3D coordinates of human body joints [30, 25, 44].

The goal of this paper is to incorporate dynamical information from each pixel in the video, to better capture human body motion for the task of activity recognition. Figure 1 shows an overview of the proposed method. The motion of the human body is represented as a function of time, which is mapped into a latent space, providing complimentary information for recognizing the activities.

Our work makes the following contributions:

- A novel dynamic encoder model that captures the temporal information of body joint movements and produces a video level representation.
- Extensive experiments on using the dynamic motion representation, called DynaMotion, and feeding them to a CNN for the task of human action classification.
- We achieve the best performance on several action recognition benchmark datasets by combining the dynamic representation with appearance and motion streams.

The paper is organized as follows: in section 2 we review the most recent works and categorize them in terms of type of their approach, while section 3 speaks about the core idea of dynamic encoding and provides background details. Section 4 shows how we implement the proposed method. Finally, in section 5 we extensively study the DynaMotion model and its varieties, evaluating recognition accuracy and comparing with state-of-the-art methods on the UCF101, JHMDB and HMDB51 datasets, as well as AVA in case of action localization.

2. Related work

Activity recognition aims to recognize common human activities in real life settings. Datasets such as UCF101 and HMDB51, collected to help this field, provide realistic videos of different persons, performing different actions in controlled settings. In the following, we categorize recent works for the task of activity recognition using four main approaches: 1) Combining multiple modalities, such as raw RGB frames, optical flow, and audio [66, 21, 70, 73] 2) Spatio-temporal convolutions and 3D convolutions [65, 59, 22, 35, 50, 64], 3) Recurrent models and Long Short-Term Memory based methods [33, 38], and finally, 4) Video Representation methods, such as [4].

Using multiple modalities: Kalogeiton et al. [28] introduced an Action Tubelet detector, which produces a sequence of bounding boxes with scores, where they used the SSD detector to extract a set of anchor cuboids. Another work [55] used a network based on bottleneck modules, where each module has two sparse coding layers with wide and slim dictionaries. [43] looked at the problem of spatio-temporal localization and classification of concurrent actions, using color images, optical flow, and motion detection scores, where they construct action tubes by solving two energy maximization problems with dynamic programming.

The work of [60] proposed a temporal segment network (based on the idea of long-range temporal structure modeling) for video-based action recognition where they combine a sparse temporal sampling strategy and video-level supervision. [46] focused on attention based modeling to find out the salient portions while capturing the long-term dependencies.

Spatio-temporal and 3D CNNs: A recent work [53] showed the importance of aggregation of temporal and spatial streams for the task of action recognition by distillation.

Wang et al. [61] proposed a pyramid network for spatial and temporal feature fusion.

3D-convolution over short video clips - typically just a few seconds - learn motion features from raw frames implicitly and then aggregate predictions at the video level. Karpathy et al. [29] demonstrated that their network is just marginally better than single frame baseline, which indicates learning motion features is difficult. In view of this, Simonyan et al. [49] directly incorporated motion information from optical flow, but only sampled up to 10 consecutive frames at inference time. The disadvantage of such local approaches is that each frame/clip may contain only a small part of the full videos information, resulting in a network that performs no better than the naive approach of classifying individual frames.

RNN and LSTM based models: Initial proposals based on deep architectures for video recognition consisted of a feature extraction block at the frame level through sequential CNNs, followed by a recurrent network, such that they were jointly trained to simultaneously learn temporal dynamics and convolutional perceptual representations

[11]. Several recent works have been inspired by this procedure, such as [67], in which spatio-temporal CNN features are extracted from video clips sliced with a fixed length so that sequential appearance and dynamic information are learned through a LSTM. In contrast, other proposed approaches are not trained end-to-end and use Bag of Words or dominant motion as pre-computed feature descriptors, followed by a LSTM-RNN [3].

Recent work by Yue-Hei Ng [68] considered several different ways to aggregate strong CNN image features over long periods of a video (tens of seconds) including feature pooling and recurrent neural networks.

In contrast, the Long Short Term Memory (LSTM) [11] uses memory cells to store, modify, and access internal state, allowing it to better discover long-range temporal relationships.

Another approach [68] incorporated five stacked layers of LSTMs after CNNs for temporal information extraction (The CNN outputs are passed upwards to the LSTM layers and forward through time). They also analyze different convolutional temporal feature pooling to better design a CNN for this task. A recent work in action recognition [58] improved dense trajectories by explicitly estimating camera motion which results in a better video representation for action recognition. They matched feature points for different frames by SURF descriptors and dense optical flow.

Video Representation: Another work on video representation for action recognition is the work of [5] where they produced a single RGB image per video by rank pooling on its raw image pixels. Dynamic images are used to summarize actions and motions happening in a video by temporal pooling as a layer in CNN. Another work [18] built a fully convolutional feed-forward auto-encoder to learn both the local features and the classifiers as an end-to-end learning framework. The auto-encoder learned the regularity dynamics in long-duration videos and can be useful for identifying irregularity in the videos (abnormal event detection). Also the low level motion features were learned using a fully convolutional auto-encoder.

Our proposed method can be also categorized as a video representation model. In this work we benefit from one of the state-of-the-art methods in object detection and instance segmentation, Mask R-CNN [19], which also provides key-point estimation for a variety of objects. We approach the problem of human activity recognition by estimating human body pose (using Mask R-CNN to detect person, as one class of object, along with its key-points and heat-maps). This method efficiently detects multiple objects (in our case, persons) in an image, while estimating human pose simultaneously, allowing us to further exploit the human body dynamics from video frames.

3. Dynamic Motion Representation

In this section, we present our video representation model, DynaMotion, to encode the human body motion. We start with the heat-map extraction in section 3.1 and then describe the dynamical encoder model in 3.2. Finally, we show the best performing method as a three-streams network consisting of RGB, optical flow, as well as the proposed DynaMotion representation as parallel streams.

3.1. Body Joint Extraction and Heat-maps

Recent advances in 2D and 3D pose estimation make it easy to obtain the coordinates of human body joints (and other objects' key-points) [6, 19]. One can also extract mid-level features and heat-maps from the networks trained on datasets with pose annotations. Heat-maps can be interpreted as an approximation to the probability of having a body joint at each pixel. Here, we use the Mask R-CNN [19] for extracting the joint heat-maps for action recognition, in addition to the person bounding boxes as part of our model for action localization. We chose this model because it detects multiple objects and their key-points (in our case, human body and joints) in a given image, provides a mask for each object, and it is robust against occlusion. In particular, we use the person bounding boxes and the joint heat-maps from a pre-trained Mask R-CNN on COCO dataset [34]. Detailed experiments and discussion on why heat-maps are useful are provided in our ablation study, part 5.2.

We pass each frame through Mask R-CNN and extract the heat-map for each key-point. The model extracts 17 body joints (5 for head and 3 for each of the 4 limbs), resulting in an output with 18 channels: one heat-map for each joint plus one channel for background. We then stack these channels on top of each other to have a combined heat-map for all the body joints. The spatial resolution of the heat-map is lower than the original frame, which we up-sample to a fixed size of 64×64. In the implementation details, we denote the size of the heat-map after re-scaling by $W \times H$. The value for each pixel in the pose heat-map is between 0 and 1, representing the probability of the corresponding pixel belonging to a specific body joint. In the following, we propose an efficient approach for encoding the temporal evolution of these heat-maps as an input to our network.

3.2. Affine Invariant Dynamic Motion Encoding

In [71] Zhang *et al*. modeled the motion of 3D human joints with linear time invariant systems and showed that this representation can be successfully used for activity recognition. Furthermore, in [2] Ayazoglu *et al*. showed that all affine 2D projections of a 3D motion trajectory modeled by a linear auto-regressive dynamical model can be represented using the same linear dynamical model. This suggests capturing the dynamics of the heat-map of the joints

to exploit viewpoint invariance. Thus, we propose to capture the dynamic information of the joints using the recently proposed DYAN [37] dynamics-based encoder-decoder network. DYAN was proposed in the context of video frame prediction, but can be applied to any temporal sequence, provided that it can be approximated by the output of a linear system and hence can be applied here. We chose this encoder because it uses very few parameters, it is easy to train and has shown excellent predictive performance, but more importantly, because the model exploits the affine viewpoint invariance described above.

During unsupervised training, DYAN learns a structured dictionary D of size $T \times N$ to encode input sequences $y_{1:T}$ of length T using a set of N dynamic-based atoms. These atoms (columns of D) are the impulse responses of low order (first and second order) linear time invariant systems, which are parameterized by the magnitude ρ_i and phase Φ_i of their poles $p_i = \rho_i e^{j\Phi_i}$:

$$D = \begin{bmatrix} 1 & 1 & \dots & 1 \\ p_1 & p_2 & \dots & p_N \\ p_1^2 & p_2^2 & \dots & p_N^2 \\ \vdots & \vdots & \dots & \vdots \\ p_1^{T-1} & p_2^{T-1} & \dots & p_N^{T-1} \end{bmatrix}$$

Then, the encoding of a sequence $y_{1:T}$ is given by a very sparse vector of coefficients c that selects and weighs the atoms in the dictionary. The vector c is found by solving an sparsification problem:

$$\min_c \frac{1}{2}\|y_{1:T} - Dc\|_2^2 + \lambda\|c\|_1$$

where the first term seeks a good fitting of the input data while the second term penalizes higher order systems. That is, the encoding seeks to explain the input data using as few as possible poles, i.e. as the output of the "simplest" linear system that fits well the input data, where "complexity" of the system is measured by the number of its poles (For more details please refer to DYAN [37]). Note that the vector c has dimension N, i.e. the number of atoms, regardless of the length of the input, and as mentioned above, it should be sparse.

3.3. Appearance and Dynamics Aggregation

We use the encoding method mentioned above to obtain a fixed size video clip representation for each input video. By training a convolutional neural network on top of the dynamic encoded video representation, the model is able to learn the dictionary D. Therefore, we can classify the action happening in a sequence of frames, given the vector of coefficients c selecting the set of atoms for each class of actions. This information focuses on the motion of the actor and it is complimentary to the context information coming from the original frames and their optical flow (as parallel RGB and OF streams). We use a pre-trained state-of-the-art model, called I3D [7], and fine-tune it for each dataset in the experiments. Finally, we aggregate the information coming from each stream to obtain the classification score for a given video clip. More details on how to merge scores coming from each of these three streams are provided in section 4.2.

4. Implementation Details

In this section we start in 4.1 with a description on how to incorporate the dynamical atom-based encoder with pose heat-maps in order to classify actions. Then, in section 4.2 we explain the network architecture for our dynamic based encoder followed by some implementation details.

4.1. Dynamic Encoding

We start by processing each frame of the input video with the Mask R-CNN model and exporting the heat-map for body joints as well as person bounding boxes. The rest of the DynaMotion method uses the person crop in order to avoid the background impact on the performance and computational cost. We stack 18 channels of the resulting heat-maps in a singe channel and then flatten a set of T consecutive $W \times H$ heat-maps into $WHT \times 1$ vectors. Then, we feed these vectors into a DYAN encoder layer [37]. The output of the dynamic encoder layer is a set of sparse WH vectors of dimension $N \times 1$, which we reshape to $W \times H \times N$ features. Thus, the encoder produces a feature vector of the same spacial size of the input ($W \times H$) with N channels (number of atoms). This layer is followed by a shallow network described in 4.2 that gives a classification score for a given video. Following [37], we define the number of atoms N to be 161 (we initialize the dictionary D with 40 poles in the first quadrant, within a ring around a unit circle, and their 3 mirror images in the other quadrants, plus a fixed pole at $p = 1$ to represent constant inputs). The overall architecture learns the dictionary D, i.e. the set of poles of the encoder layer, by minimizing a loss function that penalizes the classification error for the actions.

4.2. Network Architecture

We studied a variety of networks to train on top of the dynamic motion representation and observed that using a shallow network with six convolutional layers and one fully connected layer we can achieve the best results. Thus, the resulting architecture is a shallow network compared against to standard CNNs. We find that given the texture of our dynamic motion representation, the network does not need to be deep and can be easily trained from scratch (no pre-training). The input to the first convolution layer is the output of our encoder, which is of size $W \times H \times N$. Figure 2 shows a sketch of the architecture of our proposed network.

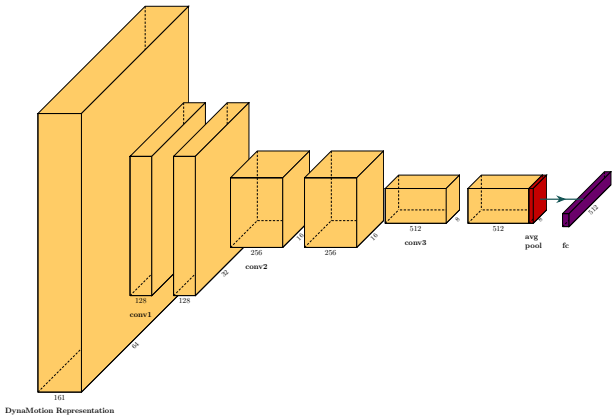

Figure 2. DynaMotion Representation Network.

The network consists of one encoding layer followed by three blocks, each block with two convolutions. The filter size for all convolution layers is 3, while the stride for the first convolution layer in each block is 2 and the second convolution in each block has stride of 1. In each block, the spacial resolution of the input is reduced, while the number of channels is doubled (first block with 128 channels and last block with 512 channels). We use batch normalization after each convolution layer followed by a ReLU. After the third block, we insert an average pooling layer, followed by a fully connected layer and softmax classifier to get the action class score. This score will be later on merged with the scores coming from the RGB and OF streams (from I3D model), resulting in a single score per frame (the merging is done by averaging the scores).

5. Ablation Study

In this section, we report extensive results evaluating the performance of action classification and localization while using the proposed dynamic motion representation on four datasets. We start by introducing the datasets used in these experiments in section 5.1, then provide the details for our dynamic encoder model and its network parameters in sections 5.2 and 5.3, respectively. In order to understand the effectiveness of our model, we provide experimental results showing the impact of using our network in section 5.4. Finally, section 5.5 shows the comparison of our best model against the state of the art on three main datasets for activity recognition and one dataset for action localization.

5.1. Datasets

For the task of activity recognition we use three main datasets (HMDB51, UCF101, and JHMDB) to examine our model's ability to learn dynamics of human body motion in different scenarios. We also use our model for the task of action localization on the AVA dataset.

HMDB51 [32] is a dataset consisting of 51 classes of actions with a minimum of 101 clips per class. This dataset has 6,849 video clips in total, from movies and YouTube, that come with pre-computed features such as HoG and STIP. We only use the video files for training our models.

Joints information for 21 classes of HMDB51 has been provided in the **JHMDB** dataset [25]. This dataset includes puppet optical flow and mask as well as joint position per frame for 928 videos. Labels for these videos are in the form of one action class per clip. There is also a meta label per clip, e.g. number of people, view point, etc. Therefore, one can use the pose annotations of this dataset to train a model for activity recognition using pose in a supervised manner.

The **UCF-101** dataset [52] has 101 action categories in three splits (about 13K video clips in total). Same as HMDB, there is one label per video clip.

AVA The Atomic Visual Actions [17] dataset consists of 80 action classes, 430 video clips in total (divided in 235 for training, 64 for validation, and 131 for testing). This dataset has localized action labels in space and time, in total 1.58M labels, as well as bounding boxes around the person involved in an action. We use version 2.1 of AVA dataset in our work.

5.2. Encoding Pose

We used the encoder layer of DYAN [37] to extract the dynamic motion representations and feed them to the CNN. Using the code provided by the authors, we set the number of poles to 40 and time horizon (number of input frames) to $T = 30$. As explained above, the underlying assumption is that human activities can be modeled as low order dynamical models, and the method learns how many atoms (i.e the order of the system) and which ones to choose from the pool of atoms. For more details please refer to part 3.2 of DYAN paper [37]. We tested using the dynamic encoder on both joint locations (coordinates) as well as joint heatmaps (as two different input types). Table 1 shows the use of heat-maps versus joint location (coordinates) as the input data type to our encoder. Our experiments show that our model performs best using heat-maps, as they convey more information per pixel.

We also trained a 3D convolution based type of network, C3D [56], on top of the heat-maps (instead of encoded heatmap, as in DynaMotion) to show the effectiveness of encoding dynamics with our model. As shown in the third row of table 1, our network outperforms the C3D network, even though both networks use temporal evolutions of pose heatmaps as their input.

5.3. Dynamic Motion CNN

In this section we study the parameters of the DynaMotion network and the impact of augmentation for training purposes. We examined our network using shallow archi-

Method	JHMDB-GT	JHMDB	HMDB	UCF101
DynaMotion with heat-maps	**69.7** %	**60.2** %	**49.1** %	**63.5** %
DynaMotion with joint coordinates	63.8 %	53.4 %	40.3 %	52.9 %
C3D [56] with heat-maps	57.7 %	37.3 %	31.5 %	44.0 %

Table 1. Mean classification accuracy using DynaMotion with heat-maps, DynaMotion with joint coordinates, and C3D with heat-maps.

Augmentation	JHMDB	HMDB	UCF101
flip (right to left)	60.2 %	49.1 %	63.5 %
no augmentation	51.4 %	46.3 %	61.9 %

Table 2. Accuracy with and without data augmentation for three different datasets.

Method	JHMDB-GT	JHMDB	HMDB	UCF101
DynaMotion	69.7 %	60.2 %	49.1 %	63.5 %
C3D [56]	56.4 %	56.4 %	51.5 %	82.1 %
C3D [56] + DynaMotion	71.3 %	69.4 %	65.3 %	93.4 %
R(2+1)D [57]	79.2 %	79.2 %	77.9 %	95.1 %
R(2+1)D [57] + DynaMotion	86.2 %	85.7 %	82.6 %	96.3 %
I3D [7]	87.0 %	87.0 %	82.1 %	97.7 %
I3D [7] + DynaMotion	**89.2** %	**87.2** %	**84.2** %	**98.4** %

Table 3. Mean classification accuracy for split 1 using combination of DynaMotion with state-of-the-art two-stream networks and spatio-temporal convolutions methods.

tectures as well as deeper networks, but the best accuracy was obtained with the six layers CNN on top of DYAN's encoder. We trained our model for 100 epochs for each dataset.

Based on our experiments, augmenting the data by flipping frames (right to left) helps increasing the mean classification accuracy. Table 2 shows the data augmentation impact on mean classification accuracy for split one of the JHMDB, HMDB51, and UCF101 datasets. The impact of data augmentation for UCF101 dataset is about 2%, while it increases the accuracy for JHMDB by almost 10%. The accuracy gain in the case of HMDB51 is about 3%. This makes sense since it is smaller than UCF101 and larger than JHMDB in terms of data size. Based on this observation, we augmented all datasets for our experiments.

5.4. Impact of the DynaMotion Representation

In order to understand the importance of using a dynamic motion representation, we compared results with and without the dynamic motion encoding. For this set of experiments we used C3D [56], R(2+1)D [57], and I3D [7] networks to combine with our dynamic motion network. We used split 1 of HMDB51, JHMDB, and UCF101 in order to evaluate the advantage of using the proposed DynaMotion representation.

Table 3 shows the mean classification accuracy to verify whether the DynaMotion representation is useful and complimentary to two stream networks as well as 3D convolutional networks. In this table, JHMDB-GT is the case where we use the ground truth puppet pose annotation to train our network instead of using Mask R-CNN to estimate the pose for each frame. The 2D annotation from JHMDB includes the x, y coordinates of each joint (15 in total) which we used to synthetically generate the pose similar to [19] joint heatmap for training. As in table 3 using the ground truth joint shows almost 9% improvements for our DynaMotion representation (first row).

The rest of the rows compare the impact of our video level representation on existing multi-stream networks as well as 3D convolutions (the original models do not use pose for training, therefore their accuracy for both JHMDB and JHMDB-GT is the same). For the purpose of this comparison, we fine-tuned C3D, R(2+1)D, and I3D for each dataset and then merged their scores with the scores coming from our model (the merging was done by averaging the scores coming from each stream and DynaMotion). Based on this table, the gain in accuracy as a result of adding DynaMotion is about 10% for the case of C3D, and up to 7% for R(2+1)D. While using the more recent model, I3D [7], we observed a slight improvement for each dataset (up to 2%), since I3D was pre-trained on Kinetics [30] which is a richer dataset. Based on these results, we conclude that DynaMotion brings complimentary information to the existing 3D convolutions and two-stream networks, and has higher impact on models with less training data.

5.5. Comparison with State-of-the-art

In this section we compare our results with the state-of-the-art in activity recognition models. Table 4 shows our results comparing to state-of-the-art methods on all splits of the three datasets (JHMDB, HMDB, and UCF101). For this experiments, we used our best model (with data augmentation and in combination with I3D), based on our previous experiments described above. We outperform all existing models (comparing to their best results using different modalities), including the models that benefit from pose [10, 75]. For JHMDB we outperform the PoTion representation model by almost 2%. On HMDB we report 84.2%, which is an improvement of almost 3% comparing to SVMP+I3D [59]. We also outperform the mean accuracy on UCF101 by a small margin (some results are shown in Figure 4 for a few randomly selected videos). Some of the models reported in this table were pre-trained on different datasets (Kinetics [7], Sports-1M [56]) using different modalities of input data, therefore the comparison might not be entirely fair. Overall, we outperform state-of-the-art models in human action recognition for JHMDB, HMDB, and UCF101. As shown in Figure 3, our model increases the per-class accuracy for most of the JHMDB classes, complementing the information coming from RGB and OF to further increase the classification accuracy with the help of

Method	JHMDB	HMDB	UCF101
CNN+hid6 [69]	-	-	79.3 %
FV+IDT [41]	-	-	84.8 %
PoseFlow [70]	-	51.74 %	-
MiCT [74]	-	63.8 %	88.9 %
P-CNN [9]	-	72.2 %	-
Chained 3D-CNN [75]	76.1 %	69.7 %	91.1 %
Attention Cluster [38]	-	69.2 %	94.6 %
CoViAR+OF [63]	-	70.2 %	94.9 %
TVNet [12]	-	72.6 %	95.4 %
OFF [54]	-	74.2 %	96.0 %
R(2+1)D [57]	-	78.7 %	97.3 %
I3D [7]	-	80.7 %	98.0 %
I3D + PoTion [10]	85.5 %	80.9 %	98.2 %
SVMP+I3D [59]	-	81.3 %	-
DynaMotion + I3D	**87.3** %	**84.2** %	**98.4** %

Table 4. Mean per-class accuracy for JHMDB, HMDB51 and UCF101 (averaged over 3 splits) in comparison with state-of-the-art.

body pose (for example in the case of golf, clap and jump, where the pose is well-defined).

We also compared our model for the task of action localization (ActivityNet challenge [66], task B). For this purpose, we used the bounding boxes extracted from Mask R-CNN model to localize subjects and used our DynaMotion representation for the cropped frames. Table 5 shows our performance in comparison with state-of-the-art on the AVA dataset [17]. We report mean average precision (mAP) of 25.8% for the validation set of AVA (for IoU=0.5). For this experiment, we used our best model results (DynaMotion+I3D) on action classification with localization results coming from Mask R-CNN [19] (as person detection bounding boxes). We used a time horizon of $T = 30$ for this experiment, having 30 frames of video as input to our DynaMotion network.

Discussion Overall, the gain in mean accuracy for the task of action recognition shows the significant impact of our DynaMotion representation. As seen in table 3, benefiting from pose and the dynamic representation adds to the power of action classification for all models, depicting the role of human body motion in addition to the context information coming from RGB and Optical Flow streams. As expected, our model performs better when the activity involves a more clear human body motion, such as *jump* or *sit*. For the classes in which the difference in human motion is negligible, our model performance is lower and therefore appearance of the subject and the context of the video has a bigger impact than pose.

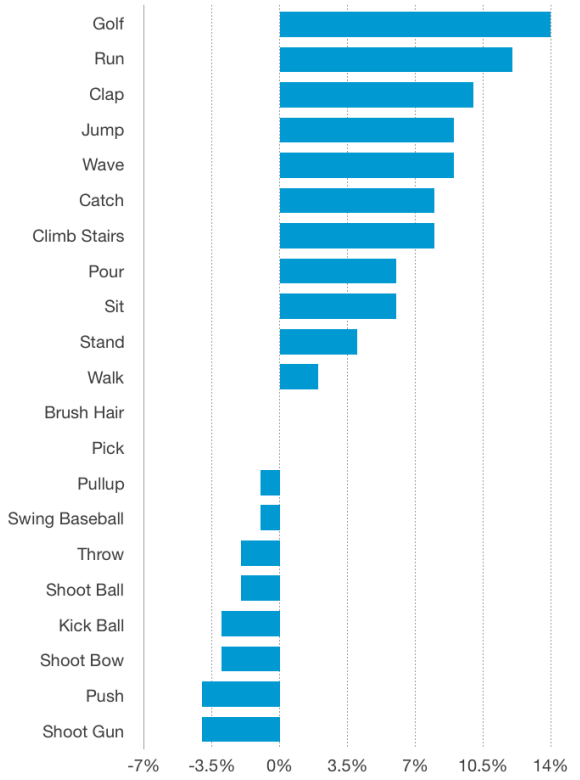

Figure 3. Accuracy improvements (per-class) for JHMDB split 1 using DynaMotion with I3D.

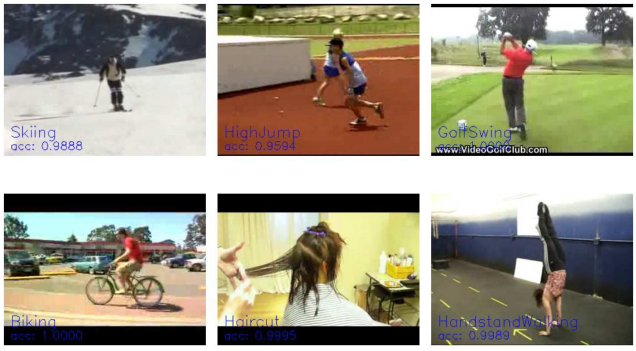

Figure 4. Action recognition results on UCF101 sample videos.

6. Conclusion

In this work we introduced Dynamic Motion Representation (DynaMotion) to encode human body motion in video clips. Using this novel video representation model, we are able to train a shallow network to classify human actions in videos. We showed that our DynaMotion representation leads to the state-of-the-art performance on UCF101, HMDB, JHMDB, and AVA datasets. As a future work, end-to-end training of the joint heat-map estimation and DynaMotion network is desired in order to study the impact

Model	Modalities	mAP@IoU0.5
AVA baseline [17]	RGB+Flow	18.4 %
Girthar et al. [14]+JFT	RGB	22.8 %
RTPR [33]	RGB+Flow	22.3 %
YH Tech [66]	RGB+Flow	22.2 %
Jiang et al. [26]	RGB+Flow	25.6 %
Ours	RGB+Flow+Pose	**25.8** %

Table 5. Per frame mean average precision for AVA validation set, using IoU=0.5

of different body joints in specific action classes.

References

[1] S. Abu-El-Haija, N. Kothari, J. Lee, P. Natsev, G. Toderici, B. Varadarajan, and S. Vijayanarasimhan. Youtube-8m: A large-scale video classification benchmark. *arXiv preprint arXiv:1609.08675*, 2016.

[2] M. Ayazoglu, B. Li, C. Dicle, M. Sznaier, and O. I. Camps. Dynamic subspace-based coordinated multicamera tracking. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2462–2469. IEEE, 2011.

[3] M. Baccouche, F. Mamalet, C. Wolf, C. Garcia, and A. Baskurt. Action classification in soccer videos with long short-term memory recurrent neural networks. In *International Conference on Artificial Neural Networks*, pages 154–159. Springer, 2010.

[4] H. Bilen, B. Fernando, E. Gavves, and A. Vedaldi. Action recognition with dynamic image networks. *arXiv preprint arXiv:1612.00738*, 2016.

[5] H. Bilen, B. Fernando, E. Gavves, A. Vedaldi, and S. Gould. Dynamic image networks for action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3034–3042, 2016.

[6] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *CVPR*, 2017.

[7] J. Carreira and A. Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, pages 4724–4733. IEEE, 2017.

[8] C. S. Catalin Ionescu, Fuxin Li. Latent structured models for human pose estimation. In *International Conference on Computer Vision*, 2011.

[9] G. Chéron, I. Laptev, and C. Schmid. P-cnn: Pose-based cnn features for action recognition. In *Proceedings of the IEEE international conference on computer vision*, pages 3218–3226, 2015.

[10] V. Choutas, P. Weinzaepfel, J. Revaud, and C. Schmid. Potion: Pose motion representation for action recognition. In *CVPR 2018*, 2018.

[11] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2625–2634, 2015.

[12] L. Fan, W. Huang, S. E. Chuang Gan, B. Gong, and J. Huang. End-to-end learning of motion representation for video understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6016–6025, 2018.

[13] B. Ghanem, J. C. Niebles, C. Snoek, F. Caba Heilbron, H. Alwassel, V. Escorcia, R. Khrisna, S. Buch, and C. Duc Dao. The activitynet large-scale activity recognition challenge 2018 summary, 2018.

[14] R. Girdhar, J. Carreira, C. Doersch, and A. Zisserman. A better baseline for ava. *arXiv preprint arXiv:1807.10066*, 2018.

[15] A. Gorban, H. Idrees, Y.-G. Jiang, A. Roshan Zamir, I. Laptev, M. Shah, and R. Sukthankar. THUMOS challenge: Action recognition with a large number of classes. http://www.thumos.info/, 2015.

[16] R. Goyal, S. E. Kahou, V. Michalski, J. Materzynska, S. Westphal, H. Kim, V. Haenel, I. Fruend, P. Yianilos, M. Mueller-Freitag, et al. The" something something" video database for learning and evaluating visual common sense.

[17] C. Gu, C. Sun, S. Vijayanarasimhan, C. Pantofaru, D. A. Ross, G. Toderici, Y. Li, S. Ricco, R. Sukthankar, C. Schmid, et al. Ava: A video dataset of spatio-temporally localized atomic visual actions. *arXiv preprint arXiv:1705.08421*, 3(4):6, 2017.

[18] M. Hasan, J. Choi, J. Neumann, A. K. Roy-Chowdhury, and L. S. Davis. Learning temporal regularity in video sequences. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 733–742, 2016.

[19] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, pages 2980–2988. IEEE, 2017.

[20] L. A. Hendricks, Z. Akata, M. Rohrbach, J. Donahue, B. Schiele, and T. Darrell. Generating visual explanations. In *European Conference on Computer Vision*, pages 3–19. Springer, 2016.

[21] O. Hommos, S. L. Pintea, P. S. Mettes, and J. C. van Gemert. Using phase instead of optical flow for action recognition. *arXiv preprint arXiv:1809.03258*, 2018.

[22] D.-A. Huang, V. Ramanathan, D. Mahajan, L. Torresani, M. Paluri, L. Fei-Fei, and J. C. Niebles. What makes a video a video: Analyzing temporal information in video understanding models and datasets. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7366–7375, 2018.

[23] H. Idrees, A. R. Zamir, Y. Jiang, A. Gorban, I. Laptev, R. Sukthankar, and M. Shah. The thumos challenge on action recognition for videos in the wild. *Computer Vision and Image Understanding*, 155:1–23, 2017.

[24] C. Ionescu, D. Papava, V. Olaru, and C. Sminchisescu. Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(7):1325–1339, jul 2014.

[25] H. Jhuang, J. Gall, S. Zuffi, C. Schmid, and M. J. Black. Towards understanding action recognition. In *International*

Conf. on Computer Vision (ICCV), pages 3192–3199, Dec. 2013.

[26] J. Jiang, Y. Cao, L. Song, S. Z. Y. Li, Z. Xu, Q. Wu, C. Gan, C. Zhang, and G. Yu. Human centric spatio-temporal action localization.

[27] J. Johnson, A. Karpathy, and L. Fei-Fei. Densecap: Fully convolutional localization networks for dense captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4565–4574, 2016.

[28] V. Kalogeiton, P. Weinzaepfel, V. Ferrari, and C. Schmid. Action tubelet detector for spatio-temporal action localization. *arXiv preprint arXiv:1705.01861*, 2017.

[29] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale video classification with convolutional neural networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 1725–1732, 2014.

[30] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017.

[31] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[32] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre. Hmdb: a large video database for human motion recognition. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2556–2563. IEEE, 2011.

[33] D. Li, Z. Qiu, Q. Dai, T. Yao, and T. Mei. Recurrent tubelet proposal and recognition networks for action detection. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 303–318, 2018.

[34] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[35] K. Liu, W. Liu, C. Gan, M. Tan, and H. Ma. T-c3d: Temporal convolutional 3d network for real-time action recognition. In *AAAI*, 2018.

[36] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia. Path aggregation network for instance segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8759–8768, 2018.

[37] W. Liu, A. Sharma, O. Camps, and M. Sznaier. Dyan: A dynamical atoms-based network for video prediction. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 170–185, 2018.

[38] X. Long, C. Gan, G. de Melo, J. Wu, X. Liu, and S. Wen. Attention clusters: Purely attention based local feature integration for video classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7834–7843, 2018.

[39] M. Marszałek, I. Laptev, and C. Schmid. Actions in context. In *IEEE Conference on Computer Vision & Pattern Recognition*, 2009.

[40] M. Monfort, A. Andonian, B. Zhou, K. Ramakrishnan, S. A. Bargal, Y. Yan, L. Brown, Q. Fan, D. Gutfreund, C. Vondrick, et al. Moments in time dataset: one million videos for event understanding. *IEEE transactions on pattern analysis and machine intelligence*, 2019.

[41] X. Peng, C. Zou, Y. Qiao, and Q. Peng. Action recognition with stacked fisher vectors. In *European Conference on Computer Vision*, pages 581–595. Springer, 2014.

[42] M. Rodriguez. Spatio-temporal maximum average correlation height templates in action recognition and video summarization. 2010.

[43] S. Saha, G. Singh, M. Sapienza, P. H. Torr, and F. Cuzzolin. Deep learning for detecting multiple space-time action tubes in videos. *arXiv preprint arXiv:1608.01529*, 2016.

[44] A. Shahroudy, J. Liu, T.-T. Ng, and G. Wang. Ntu rgb+ d: A large scale dataset for 3d human activity analysis. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1010–1019, 2016.

[45] A. Shahroudy, J. Liu, T.-T. Ng, and G. Wang. Ntu rgb+d: A large scale dataset for 3d human activity analysis. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[46] Y. Shi, Y. Tian, Y. Wang, and T. Huang. Joint network based attention for action recognition. *arXiv preprint arXiv:1611.05215*, 2016.

[47] G. A. Sigurdsson, A. Gupta, C. Schmid, A. Farhadi, and K. Alahari. Actor and observer: Joint modeling of first and third-person videos. In *CVPR*, 2018.

[48] G. A. Sigurdsson, G. Varol, X. Wang, A. Farhadi, I. Laptev, and A. Gupta. Hollywood in homes: Crowdsourcing data collection for activity understanding. In *European Conference on Computer Vision*, 2016.

[49] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in neural information processing systems*, pages 568–576, 2014.

[50] G. Singh, S. Saha, and F. Cuzzolin. Predicting action tubes. *arXiv preprint arXiv:1808.07712*, 2018.

[51] K. Soomro and A. R. Zamir. Action recognition in realistic sports videos. In *Computer vision in sports*, pages 181–208. Springer, 2014.

[52] K. Soomro, A. R. Zamir, and M. Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012.

[53] J. C. Stroud, D. A. Ross, C. Sun, J. Deng, and R. Sukthankar. D3d: Distilled 3d networks for video action recognition. *arXiv preprint arXiv:1812.08249*, 2018.

[54] S. Sun, Z. Kuang, L. Sheng, W. Ouyang, and W. Zhang. Optical flow guided feature: a fast and robust motion representation for video action recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[55] X. Sun, N. M. Nasrabadi, and T. D. Tran. Supervised multilayer sparse coding networks for image classification. *arXiv preprint arXiv:1701.08349*, 2017.

[56] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4489–4497, 2015.

[57] D. Tran, H. Wang, L. Torresani, J. Ray, Y. LeCun, and M. Paluri. A closer look at spatiotemporal convolutions for action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6450–6459, 2018.

[58] H. Wang and C. Schmid. Action recognition with improved trajectories. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3551–3558, 2013.

[59] J. Wang, A. Cherian, F. Porikli, and S. Gould. Video representation learning using discriminative pooling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1149–1158, 2018.

[60] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. Van Gool. Temporal segment networks: towards good practices for deep action recognition. In *European Conference on Computer Vision*, pages 20–36. Springer, 2016.

[61] Y. Wang, M. Long, J. Wang, and P. S. Yu. Spatiotemporal pyramid network for video action recognition. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 1529–1538, 2017.

[62] P. Weinzaepfel, X. Martin, and C. Schmid. Human action localization with sparse spatial supervision. *arXiv preprint arXiv:1605.05197*, 2016.

[63] C.-Y. Wu, M. Zaheer, H. Hu, R. Manmatha, A. J. Smola, and P. Krähenbühl. Compressed video action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6026–6035, 2018.

[64] S. Xie, C. Sun, J. Huang, Z. Tu, and K. Murphy. Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 305–321, 2018.

[65] S. Yan, Y. Xiong, and D. Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. *arXiv preprint arXiv:1801.07455*, 2018.

[66] T. Yao and X. Li. Yh technologies at activitynet challenge 2018. *arXiv preprint arXiv:1807.00686*, 2018.

[67] Y. Ye and Y. Tian. Embedding sequential information into spatiotemporal features for action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 37–45, 2016.

[68] J. Yue-Hei Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and G. Toderici. Beyond short snippets: Deep networks for video classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4694–4702, 2015.

[69] S. Zha, F. Luisier, W. Andrews, N. Srivastava, and R. Salakhutdinov. Exploiting image-trained cnn architectures for unconstrained video classification. *arXiv preprint arXiv:1503.04144*, 2015.

[70] D. Zhang, G. Guo, D. Huang, and J. Han. Poseflow: A deep motion representation for understanding human behaviors in videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6762–6770, 2018.

[71] X. Zhang, Y. Wang, M. Gou, M. Sznaier, and O. Camps. Efficient temporal sequence comparison and classification using gram matrix embeddings on a riemannian manifold. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4498–4507, 2016.

[72] H. Zhao, Z. Yan, L. Torresani, and A. Torralba. Hacs: Human action clips and segments dataset for recognition and temporal localization. *arXiv preprint arXiv:1712.09374*, 2019.

[73] Y. Zhao, Y. Xiong, and D. Lin. Recognize actions by disentangling components of dynamics. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6566–6575, 2018.

[74] Y. Zhou, X. Sun, Z.-J. Zha, and W. Zeng. Mict: Mixed 3d/2d convolutional tube for human action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 449–458, 2018.

[75] M. Zolfaghari, G. L. Oliveira, N. Sedaghat, and T. Brox. Chained multi-stream networks exploiting pose, motion, and appearance for action classification and detection. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, pages 2923–2932. IEEE, 2017.