16th International Conference on Machine Vision Applications (MVA)
National Olympics Memorial Youth Center, Tokyo, Japan, May 27-31, 2019.

Skip-Pose Vectors: Pose-based motion embedding using Encoder-Decoder models

Yuta Shirakawa Tatsuo Kozakaya
Corporate Research and Development Center, Toshiba Corporation
{yuta1.shirakawa, tatsuo.kozakaya}@toshiba.co.jp

Abstract

This paper proposes a pose-based unsupervised embedding learning method for action recognition. To classify human action based on the similarity of motions, it is important to establish a good feature space such that similar motions are mapped to similar vector representations. On the other hand, learning a feature space with this property with a supervised approach requires huge training samples, tailored supervised keypoints, and action categories. Although the labeling cost of keypoints is decreasing day by day with improvement of 2D pose estimation methods, labeling video category is still problematic work due to the variety of categories, ambiguity and variations of videos. To avoid the need for such expensive category labeling, following the success of "Skip-Thought Vectors", an unsupervised approach to model the similarity of sentences, we apply its idea to contiguous pose sequences to learn feature representations for measuring motion similarities. Thanks to handling human action as 2D poses instead of images, the model size can be small and easy to handle, and we can augment the training data by projecting 3D motion capture data to 2D. Through evaluation on the JHMDB dataset, we explore various design choices, such as whether to handle the actions as a sequence of poses or as a sequence of images. Our approach leverages pose sequences from 3D motion capture and improves its performance as much as 61.6% on JHMDB.

1 Introduction

Understanding human action plays an important role for a wide range of social and industrial scenes, such as similar video retrieval from web, analyzing customer behavior to maximize sales, and so on.

The field of human action recognition has advanced rapidly over the past few years. It has moved from hand-crafted features[1] to learned convolutional neural network features[2] and also from encoding appearance information to encoding motion information[3].

However, in the case of using such robust and accurate action classifiers in real world, several problems remain to be addressed. Firstly, these approaches often rely on large-amount of training videos for each target domain at the training phase. Secondly, we have to prepare large number of expensive annotated samples for each action category. Although several large-scale video datasets have been proposed like ActivityNet, and Kinetics, it is practically infeasible and extremely costly to prepare the same amount of videos for each target domain. Finally, as a usability problem, since these classifiers estimate input into seen categories, we cannot add unseen category to the set of output options.

This paper proposes a pose-based unsupervised motion embedding learning method and a domain-independent framework for action recognition. Based on the idea of "Skip-Thought Vectors"[4], an unsupervised representation learning method in natural language processing domain, our proposed method, "Skip-Pose Vectors", applies an encoder-decoder model to the contiguous human pose sequences. Although preparing keypoint is expensive task, thanks to improvement of 2D pose estimation methods[5] and pose tracking methods, its cost is decreasing day by day. Treating human action as a sequence of pose introduces several advantages. Firstly, it simplifies problem to solve and makes model small. Secondly, using only pose makes the model focus to motion information. Finally, this also enables us to augment the training data with samples from completely different domains, i.e. 3D motion capture data. After training, we use the encoder of our model as a motion feature extractor, apply simple dynamic time warping[6] to calculate similarities, and classify inputs with nearest neighbor method. The proposed framework is illustrated in Fig. 1.

The main contributions of this paper can be summarized as follows:

- Inspired by skip-thought vectors, this paper proposes an unsupervised motion embedding method for action recognition.
- Handling human actions as 2D pose sequences, our method allows us to augment the training data with samples from another domain.
- As combining the proposed method with simple dynamic time warping and nearest neighbor, our framework has flexibility to add unseen categories.

We evaluate our proposed method on JHMDB[7], a public datasets for action recognition tasks. Augmentation using 3D motion capture data improves the accuracy from 61.0% to 61.6% and exceeds the baseline approaches by 6.1 percentage points.

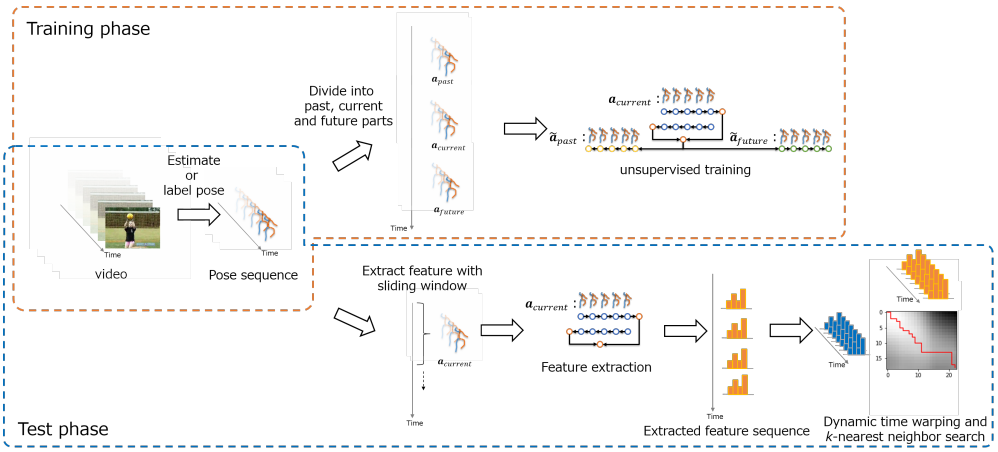


Figure 1. The area surrounded by a broken orange line shows the procedure for the training phase. The area surrounded by a broken blue line shows the procedure in the test phase. Since our approach uses only pose sequences from videos, the training phase does not need any labels for the action categories in videos.

2 Related works

2.1 Skip-Thought Vectors

Skip-thought vectors[4] is an unsupervised learning method for sentence embeddings. This method uses a sequence-to-sequence (seq2seq) model (see Fig. 2) with one encoder model $f(\cdot;\theta_c)$ and two decoder models $g(\cdot;\psi_p), g(\cdot;\psi_f)$ where θ_c, ψ_p and ψ_f are the parameters of each sub-models. Each sub-model consists of recurrent neural networks (RNNs) and whole model aims to directly model the conditional probability $p(\boldsymbol{s}_p, \boldsymbol{s}_f|\boldsymbol{s}_c)$ of mapping a certain sentence $\boldsymbol{s}_c$ into adjacent sentences $\boldsymbol{s}_f$ and $\boldsymbol{s}_p$. It accomplishes such goal through the encoder-decoder framework[8].

An important idea of this approach is that one can learn representations from co-occurrence and context of three consecutive sentences.

After training, the encoder is used to embed a sentence into a feature space and the embedded features are applied to similar sentence retrieval.

2.2 Using the Context of Pose Sequences

Since motion is a key part of actions, action recognition studies have paid special attention to modeling representations of human motion.

Martinez et al. proposed a seq2seq model for pose prediction tasks[9]. Using 3D coordinate pose sequences as inputs, they estimated future pose sequences based on the current pose sequence. From their work, it turns out that there is a contextual relationship between pose sequences as well as sentences. Srivastava et al. proposed an unsupervised pre-training approach for video recognition [10]. They encode the current sequence into feature vectors and simultaneously reconstruct the current sequence and future sequences based on encoded features. After unsupervised training, they use the trained parameter as the initial parameter of a multi-class classifier and fine-tune the parameters using a few labeled samples in a supervised manner.

2.3 Dealing with Varying Temporal Durations

One of the main issues in action recognition is that sequences representing the same action may have different lengths due to the velocity and style with which the action is performed. Various approaches are used to account for this problem, such as adopting global feature representation of the entire sequence, which generally sacrifices information about the temporal structure of the sequence. Another work has focused on feature extraction from short periods and applied dynamic time warping to calculate elastic distances[11].

3 Proposed Method

3.1 Skip-Pose Vectors

Inspired by skip-thought vectors[4], we use the seq2seq structure model shown in Fig. 2 and treat human actions as contiguous sequences. Each human action is represented with a sequence of 2D keypoints of human at each frame in the video. Specifically, let an action $\boldsymbol{a} = (\boldsymbol{P}_0 \dots \boldsymbol{P}_{T-1})$ be a sequence of poses, where $\boldsymbol{P}_t = \boldsymbol{p}_{t,0}, \dots, \boldsymbol{p}_{t,J-1}$ denotes a concatenated keypoint locations, J is the number of joints, and T is the length of a video. For each keypoint $j = 0, \dots, J-1, \boldsymbol{p}_{t,j} = (x_{t,j}, y_{t,j}), \forall (x_{t,j}, y_{t,j}) \in \mathbb{R}^2$.

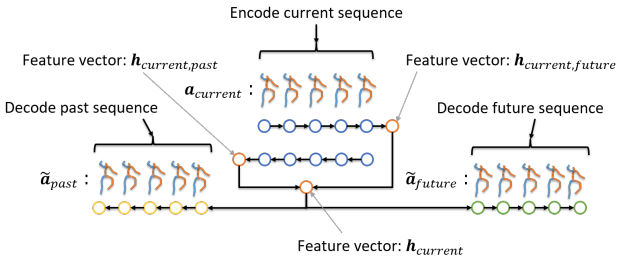
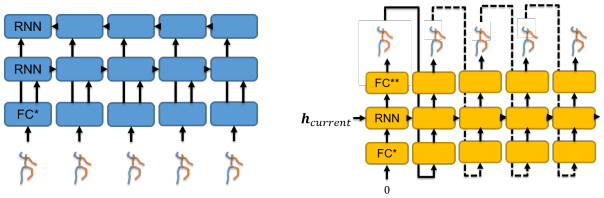


Figure 2. The skip-pose model. Given a triplets of contiguous pose sequences $(\boldsymbol{a}_p, \boldsymbol{a}_c, \boldsymbol{a}_f)$, where $\boldsymbol{a}$ is a pose sequence of fixed length T, the model encodes the current sequence $\boldsymbol{a}_c$ into feature vector $\boldsymbol{h}_c$ and tries to reconstruct the past sequence $\boldsymbol{a}_p$ and the future sequence $\boldsymbol{a}_f$ based on $\boldsymbol{h}_c$.



(a) Encoder (b) Decoder

Figure 3. The networks, which constitute our proposed method. The figure on left side shows the encoder, which treats each pose with fully-connected (FC) layer and encodes with bi-directional manner. The figure on right side shows decoder, which reconstructs the past (future) sequence based on encoded representation h_c and zero inputs. Like encoder input, decoder also use FC layer to reconstruct poses. The parameters of FC* and FC** are shared across the encoder and the two decoders.

Unfortunately, keypoints may not be visible in the images, or may not be detected even when visible. Therefore, we introduce a binary mask for each keypoint to handle these uncertainties. Let $\hat{\boldsymbol{a}} = (\boldsymbol{M}_0 \dots \boldsymbol{M}_{T-1})$ be a sequence of masks, where $\boldsymbol{M}_t = \boldsymbol{m}_{t,0}, \dots, \boldsymbol{m}_{t,J-1}$ denotes a concatenated boolean values whether corresponding keypoints are detected or not. For each mask $j = 0, \dots, J-1, \boldsymbol{m}_{t,j} = (b_{t,j}, b_{t,j}), \forall b_{t,j} \in \{0, 1\}$. We apply the mask $\hat{\boldsymbol{a}}$ to the input $\boldsymbol{a}$ with element-wise multiplication in both training and testing phase. This can be interpreted that the probability of dropout in ordinal deep learning changing to the probability of the keypoint existing outside of the frame or the probability of failed detection.

In the original skip-thought vectors, sentences delimited by periods were used as inputs and outputs, but in pose sequence there is no clear division criterion, so we fix the lengths of inputs and outputs to T.

Training is performed by minimizing the Euclidean distance between GT poses and reconstructed poses.

To simplify the model structure, we share the parameters of FC* and FC** across encoder and two decoders as denoted in Fig. 3.

$$\boldsymbol{h}_{c,f}, \boldsymbol{h}_{c,p} = f(\boldsymbol{a}_c, \boldsymbol{M}_c; \theta_c) \tag{1}$$

$$\boldsymbol{h}_c = \frac{1}{2}(\boldsymbol{h}_{c,f} + \boldsymbol{h}_{c,p}) \tag{2}$$

$$\tilde{\boldsymbol{a}}_p = g(\boldsymbol{h}_c; \psi_p) \tag{3}$$

$$\tilde{\boldsymbol{a}}_f = g(\boldsymbol{h}_c; \psi_f) \tag{4}$$

$$\hat{\theta}_c, \hat{\psi}_p, \hat{\psi}_f = \arg\min_{\theta_c, \psi_p, \psi_f} \frac{1}{4}(||\tilde{\boldsymbol{a}}_p - \boldsymbol{a}_p||^2 + ||\tilde{\boldsymbol{a}}_f - \boldsymbol{a}_f||^2) \tag{5}$$

After training, the encoder is used as a feature extractor, and the internal state after encoding the fixed-length input is used as the feature representation of the input. After extracting the feature vectors, the input and the feature sequence of each registered dictionary are aligned by dynamic programming, the similarity is calculated, and the input is classified by nearest neighbor method as described in Sec. 3.2.

3.2 Dynamic Time Warping and Similarity

After extracting the features from moments, we align the feature sequences by dynamic time warping (DTW)[6]. DTW is a simple dynamic programming algorithm that provides the best alignment between two sequences.In our case, once the similarity score map is defined by the cosine similarities of all combination of time steps between a query and a dictionary, DTW provides the optimal time alignment. For example the similarity score map $S(\cdot, \cdot)$ between query sample q_i of length T_{q_i} and dictionary sample d_j of length T_{d_j} is calculated as following, where $\boldsymbol{h}_{q_i,t_l}$ is a feature vector extracted from sample q_i at time step t_l.

$$S(q_i, d_j) = \begin{pmatrix} s_{t_0,t_0} & \cdots & s_{t_0,T_{d_j}-1} \\ \vdots & \ddots & \vdots \\ s_{T_{q_i}-1,t_0} & \cdots & s_{T_{q_i}-1,T_{d_j}-1} \end{pmatrix} \tag{6}$$

$$\text{where } s_{t_l,t_m} = cos(\boldsymbol{h}_{q_i,t_l}, \boldsymbol{h}_{d_j,t_m})$$

After obtaining the optimal time warping path P, the final similarity score between the query and the dictionary is calculated by simply summing the corresponding similarity scores.

$$score = \sum_{t_l,t_m \in P} s_{t_l,t_m} \tag{7}$$

Figure 4 shows an example of a similarity score map and the optimal time warping path.

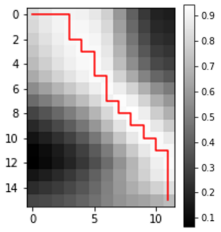
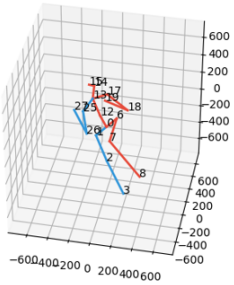
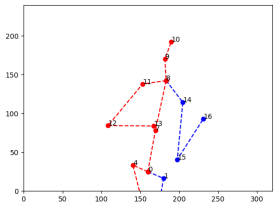


Figure 4. Example of similarity score map. Time steps in the query and dictionary are shown vertically and horizontally, respectively. Whiter colors indicates higher similarity between time steps. The red line shows the optimum path obtained by dynamic time warping.



(a) Original 3D pose



(b) Projected 2D pose

Figure 5. (a) An example of GT 3D pose. (b) Example of projected from results from certain distance. We change the camera position and obtain countless projection results.

3.3 Data Augmentation

Thanks to the recent development of pose estimation methods[5], it is possible to collect pose sequences as time series corresponding to a video image at low cost. However, automatic detection is not perfect, and the output includes false positions and misses some true positions. Therefore, we augment the training data using 3D motion capture data that is taken in advance.

By using the 3D motion capture data, it is possible to prepare countless pose sequences from arbitrary viewpoints. Furthermore, by using pose sequences other than the target domain, it can be expected to gain higher generalization performance.

Figure 5 shows an example of actor in the 3D motion capture database Human 3.6M[12], which is used in our experiments. In addition, the original 3D pose and 2D projected poses used in our augmentation are also shown in Fig. 5.

Figure 6. Some sample frames from JHMDB[7]. We overlay GT pose with RGB lines. R: the backbone; G: right side body; B: left side body. Some GT are misaligned.

4 Experiments

4.1 Settings

We evaluate our proposed method on a publicly available action recognition dataset, JHMDB[7]. JHMDB is collected from various sources such as web videos and movies, and proved to be realistic and challenging. It contains 21 categories such as “brush hair”, “catch”, “clap”, and “wave” comprising 928 videos.

A remarkable aspect of this dataset is that each video sample has been annotated with the positions of 15 body keypoints at each time step. Some example frames and GT poses are shown in Fig. 6. Thanks to this feature, we can ignore the accuracy of keypoint detection and compare the feasibility of feature extractors.

In the training phase, only keypoint annotations of training samples are used. In the evaluation phase, the training samples are registered as dictionary samples.Following the official evaluation protocol, we evaluate the mean average precision (mAP) of official cross validations.

Each video contained in JHMDB has a resolution of 240×320 px, but there is large variation in the scale and the position at which the main actor appears. In order to suppress the effect of such fluctuations, for each series $\boldsymbol{a}$, coefficients for normalizing the maximum and minimum value in the x and y direction of the keypoints position of the beginning pose $\boldsymbol{P}_0$ into the range $[-1, 1]$ is used as preprocessing to normalize the position and scale of the entire pose. As a result, the position and the scale at the start time are aligned while keeping changes in the position and scale of the person over a video. In JHMDB, however the keypoints outside the frame are also annotated, but we treat them as not detected and mask the corresponding $\boldsymbol{m}_{t,j}$ with $(0, 0)$.

To the best of our knowledge, although some previous works use supervised approaches to JHMDB, no prior works use an unsupervised approach. Therefore, we compare the following simple approaches as baseline methods.

1. Compressing the above features into 64 dimensions by PCA.

Table 1. Evaluation of different approaches and settings when applied to JHMDB showing mAP.

Method	Data augmentation	Past decoding part	Length of input/output T		
			2	4	6
PCA			55.6	55.5	55.2
AE		✓	58.7	58.5	55.4
Proposed			59.1	56.7	54.6
		✓	**60.9**	61.0	57.6
	✓	✓	60.7	**61.6**	**61.1**
Proposed with image		✓	12.8	11.7	14.0

2. Out model without RNN (i.e. replace the RNNs in proposed method to auto encoder), and treating all the times of the input sequence as a concatenated one. We keep hyperparameters such as number of layers and hidden units.

We also change several factors in the proposed method to investigate the respective effects.

1. Presence or absence of decoding the past actions.

2. Data augmentation with external 3D data.

3. Proposed method using images instead of key points. (i.e. instead of a keypoint, use an image sequence cropped with bounding box of given keypoints and resized to 64x32 are treated as input series.)

When we use RNN or auto encoder (AE) as a feature extractor, we use models with hidden units are 64 dimension.

We train our model using Adam with learning rate of 10^{-3} and regularize the model with weight decay of 10^{-5}. We use the validation set to determine the optimal training epochs

4.2 Evaluation Results

Table 1 shows the results of comparing the accuracy by changing the length T of the input sequence used for training and test. Since our approach shows higher performance than PCA, it can extract more distinctive feature vectors, better reflecting the similarity between actions, by unsupervised learning. By comparing with the results from AE, we can see the effectiveness of handling actions as pose sequences by RNNs. Remove the past decoding part from our proposed method causes it to deteriorate to the same accuracy as AE, so we can see that the past decoding part contributes positively to the performance. By adding augmented data, the accuracy is improved on the condition of the sequence length is 5 or more. This is thought to be because the variation of pose sequences does not increase even if augmented data is added to short sequences.

We also conducted an experiment of proposed method with image sequences. As Table 1 shows, this condition showed lower performance than any other baselines. It seems that JHMDB is too small to learn the context of the image sequences, or that the RNN model we used was too small to learn embeddings from images. Skip-Thought is a simple embedding learning method which is possible because it uses pose sequences.

Table 2. Comparison of mAP with state of the art approaches on JHMDB.

Method	mAP
P-CNN[13]	74.6
HLPF[7]	77.8
Proposed	61.6

Table 2 shows a comparison of the results with those of previous works. We mention that the other methods use supervised approaches whereas our method uses an unsupervised approach and shows results that are not bad.

Figure 7 shows the relationship between the amount of training data and accuracy. These results show that the accuracy increases as the training data are augmented, but the performance decreases after the augmentation data exceed 10^5 samples.

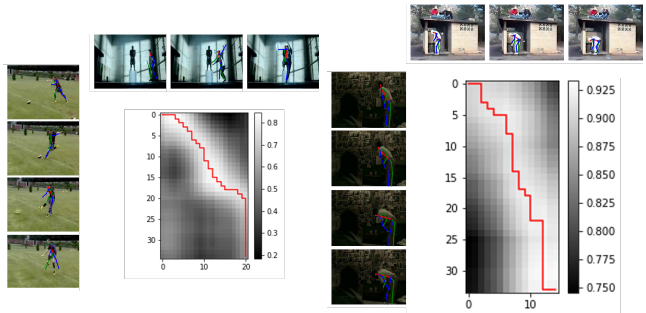
Figure 8 shows the relationship between the proportion of dictionary data to be registered and the accuracy of each method under non-augmented conditions. As the proportion of registrations changes, our proposed method consistently shows higher performance. Additionally, even when the proportion of registrations is reduced to about half, the accuracy is equivalent to the baseline methods of condition under the fully registered.

4.3 Visualization

In order to better illustrate how our method recognizes similarities between pose sequences, we provide some examples of similarity score map between the query and the nearest dictionary sample, and some thumbnails. The map shows a gradual peak and DTW tracks the peak.

5 Conclusion

In this paper, we have proposed a pose-based unsupervised motion embedding method for action recognition. Since the approach handled human motion as 2D pose sequence, we could train simple encoder-decoder model for motion embeddings and simply augment the



(a) success case (b) failed case

Figure 9. Qualitative results. In each figure, the images on the left side of the similarity map show frames from the input query and the upper side shows frames from the nearest dictionary sample.

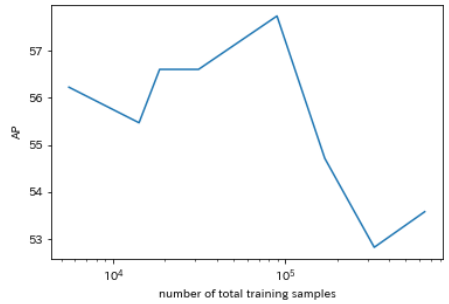


Figure 7. The relationship between the number of training samples vs. AP on JHMDB-split1. The left end of the plot shows the AP on the condition without data augmentation.

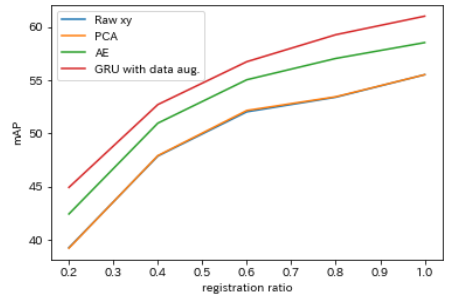


Figure 8. The relationships between the ratio of registered dictionary samples vs. mAP on JHMDB. Registration ratio = 1.0 means that all training samples are registered as dictionary samples.

training data by projecting 3D motion capture data to 2D. We have also shown that combining our proposed method with dynamic time warping and nearest neighbours, we can calculate the similarity between various length inputs and registered samples, and classify them. Experimental results on JHMDB showed that our proposed method constantly outperformed the baseline approaches. Moreover, it has shown that the accuracy was improved as much as 61.6%, which exceeded the baselines by 6.1 percentage points, by augmenting the training data with samples from completely different domains, i.e. 3D motion capture data.

References

[1] Heng Wang and Cordelia Schmid. Action Recognition with Improved Trajectories. *ICCV*, 2013. 1

[2] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale Video Classification with Convolutional Neural Networks. *CVPR*, 2014. 1

[3] Gunnar A. Sigurdsson, Santosh Divvala, Ali Farhadi, and Abhinav Gupta. Asynchronous temporal fields for action recognition. *CVPR*, 2017. 1

[4] Ryan Kiros, Yukun Zhu, Ruslan Salakhutdinov, Richard S. Zemel, Antonio Torralba, Raquel Urtasun, and Sanja Fidler. Skip-Thought Vectors. *NIPS*, 2015. 1, 2

[5] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. *CVPR*, 2017. 1, 4

[6] Meinard Müller. *Information Retrieval for Music and Motion*. Springer Berlin Heidelberg, 2007. 1, 3

[7] Hueihan Jhuang, Juergen Gall, Silvia Zuffi, Cordelia Schmid, and Michael J. Black. Towards understanding action recognition. *ICCV*, 2013. 1, 4, 5

[8] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to Sequence Learning with Neural Networks. *NIPS*, 2014. 2

[9] Julieta Martinez, Michael J. Black, and Javier Romero. On human motion prediction using recurrent neural networks. *CVPR*, 2017. 2

[10] Nitish Srivastava, Mansimov Elman, and Ruslan Salakhutdinov. Unsupervised Learning of Video Representations using LSTMs. *ICML*, 2015. 2

[11] Alexandros Andre Chaaraoui, José Ramón Padilla-López, and Francisco Flórez-Revuelta. Fusion of skeletal and silhouette-based features for human action recognition with RGB-D devices. *ICCV*, 2013. 2

[12] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3.6M: Large Scale Datasets and Predictive Methods for 3D Human Sensing in Natural Environments. *IEEE TPAMI*, 2014. 4

[13] Guilhem Chéron, Ivan Laptev, Guilhem Cheron, Ivan Laptev, and Cordelia Schmid. P-CNN: Pose-based CNN features for action recognition. *ICCV*, 2015. 5