Skeleton-DML: Deep Metric Learning for Skeleton-Based One-Shot Action Recognition

Raphael Memmesheimer Simon Häring Nick Theisen Dietrich Paulus

University of Koblenz-Landau
Active Vision Group
{raphael, simonhaering, nicktheisen, paulus}@uni-koblenz.de

Abstract

One-shot action recognition allows the recognition of human-performed actions with only a single training example. This can influence human-robot-interaction positively by enabling the robot to react to previously unseen behaviour. We formulate the one-shot action recognition problem as a deep metric learning problem and propose a novel image-based skeleton representation that performs well in a metric learning setting. Therefore, we train a model that projects the image representations into an embedding space. In embedding space similar actions have a low euclidean distance while dissimilar actions have a higher distance. The one-shot action recognition problem becomes a nearest-neighbor search in a set of activity reference samples. We evaluate the performance of our proposed representation against a variety of other skeleton-based image representations. In addition we present an ablation study that shows the influence of different embedding vector sizes, losses and augmentation. Our approach lifts the state-of-the-art by 3.3% for the one-shot action recognition protocol on the NTU RGB+D 120 dataset under a comparable training setup. With additional augmentation our result improved over 7.7%.

1. Introduction

Action recognition is a research topic that is applicable in many fields like surveillance, human robot interaction or in health care scenarios. In the past, a strong research focus was laid on the recognition of known activities, whereas learning to recognize from few samples gained popularity only recently [12, 18]. Because of RGB-D cameras availability and wide mobile indoor applicability, indoor robot systems are often equipped with them [21, 31]. RGB-D cameras that support the OpenNI SDK not only provide color and depth streams, but also provide human pose estimates in the form of skeleton sequences. These skeleton estimates allow a wide variety of higher-level applications without investing in the human pose estimation problem. As the pose estimation approach is based on depth streams [33], it is robust against background information as well as different lighting conditions and therefore also remains functional in dark environments. Especially in a robotics context, one-shot action recognition enables a huge variety of applications to improve the human-robot-interaction. A robot could initiate a dialog, when recognizing an activity that it is unfamiliar with, in order to assign a robot-behavior to the observation. This can be done with a single reference sample, while standard action recognition approaches can only recognize actions that were given during training time. In our proposed one-shot action recognition approach, observations are projected to an embedding space in which similar actions have a low distance and dissimilar actions have a high distance.

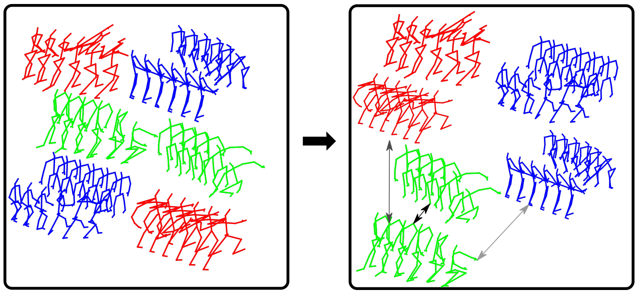

Figure 1. Illustrative example of our method. Prior to training a metric on the initial data, no class association could be formed given a skeleton sequence. After training our one-shot action recognition model, skeleton sequences can be encoded. A euclidean distance on the encoded sequence allows class association by finding the nearest neighbor in embedding space from a set of reference samples. The colors are encoding the following classes: throw, falling, grab other person's stuff. Brighter arrow colors denote higher distance in embedding space.

A high distance to all known activities can be seen as an indicator for anomalies. The embedding in a metric learning setting allows online association of novel observations, which is a high advantage over classification tasks that would require retraining or fine-tuning.

Deep metric learning based approaches are popular for image ranking or clustering, like face- or person reidentification [25, 30]. They have proven to integrate well as an association metric, e.g. in person tracking settings to reduce the amount of id-switches [30]. Even though there are skeleton-based image representations for recognizing activities from skeleton sequences, they have only recently been used to learn a metric for one-shot action recognition [18]. Fig. 1 shows an illustrative example of an application of our approach.

The contributions of this paper are as follows:

- We present a representation that reassembles skeleton sequences into images.
- We integrate the representation into a deep metric learning formulation to tackle the one-shot action recognition problem.
- We furthermore provide an evaluation of related skeleton-based image representations for one-shot action recognition.
- The source code to reproduce the results of this paper is made publicly available under `https://github.com/raphaelmemmesheimer/skeleton-dml`.

2. Related Work

Action recognition is a broad research topic that varies not only in different modalities like image sequences, skeleton sequences, data by inertial measurement units but also by their evaluation protocols. Most common protocols are cross-view or cross-subject. More recently, one-shot protocols have gained attention. As our approach focuses on skeleton-based one-shot action recognition, we present related work from the current research state directly related to our method. Skeleton based action recognition gained attention with the release of the Microsoft Kinect RGB-D camera. This RGB-D camera not only streamed depth and color images, but the SDK also supported the extraction of skeleton data. With the *NTU RGB+D* dataset [26, 12] a large scale RGB-D action recognition dataset that also contains skeleton sequences has been released. The progress made on this dataset gives a good indication of the performance of various skeleton-based action recognition approaches.

Because convolution neural architectures showed great performance in the image-classification domain, a variety of research concentrated on finding image-like representations for different research areas like speech recognition [6].

Representations for encoding spatio-temporal information were explored in-depth for recognizing actions [16, 1]. They focus on a classification context by associating class labels with skeleton sequences, in contrast to learning an embedding space. The idea of representing motion in image-like representations lead to serious alternatives to sequence classification approaches based on *Recurrent Neural Networks* [8] and *Long Short Term Memory (LSTM)* [13]. Wang et al. [28] presented joint trajectory maps. Viewpoints from each axis were set and encoded 3D trajectories for each of the three main axis views. A simple Convolutional Neural Network (CNN) architecture was used to train a classifier analyzing the joint trajectory maps. Occlusion could not be directly tackled, therefore the representation by Liu et al. [16] added flexibility by fusing up to nine representation schemes in separate image channels. A similar representation has recently shown to be usable also for action recognition on different modalities and their fusion [17]. Kim et al. [9] on the other hand presented a compact and human-interpretable representation. Joint movement contributions over time can be interpreted. Interesting to note is also the skeleton transformer by Li et al. [10]. They employ a fully connected layer to transform skeleton sequences into a 2 dimensional matrix representation.

Yang et al. [32] present a joint order that puts joints closer together if their respective body parts are connected. It is generated by a depth-first tree traversal of the skeleton starting in the lower chest. Skepxels are small 5×5-pixel segments containing the positions of all 25 skeleton joints in a random but fixed order. Liu et al. [11] use this 2D structure as it is more easily captured by CNNs. Each sample of a sequence is turned into multiple sufficiently different Skepxels which are then stacked on top of each other. These Skepxels differ only in their joint permutation. The full Skepxel-image of a sequence of skeletons is assembled width-wise, without altering the joint permutation within one row of Skepxels. Caetano et al. [1] generate two images containing motion information in the form of an orientation and a magnitude. The orientation is defined by the angles between the motion vector and the coordinate axes. The angles are stored in the color channels of an image, with time in horizontal and the joints in TSSI order in vertical direction. The gray-scale magnitude image contains the euclidean norm of the motion vectors instead.

One-shot recognition in general aims at finding a method to classify new instances with a single reference sample. Possible approaches for solving problems of this category are metric learning [27, 7], or meta-learning [4]. In action recognition, this means a novel action can be learned with a single reference demonstration of the action. In contrast to one-shot image classification, actions consist of sequential

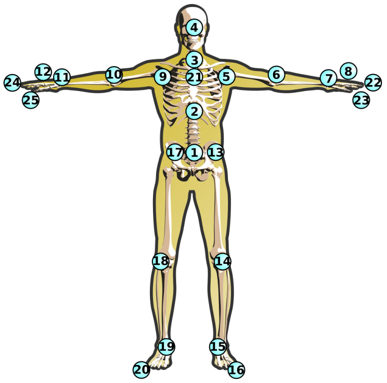

Figure 2. *NTU RGB+D 120* skeleton joint positions.

data. A single frame might not contain enough context to recognize a novel activity.

Along with the *NTU RGB+D 120* dataset, Liu et al. [12] presented a one-shot action recognition protocol and corresponding baseline approaches. The *Advanced Parts Semantic Relevance (APSR)* approach extracts features by using a spatio-temporal LSTM method. They propose a semantic relevance measurement similar to word embeddings. Body parts are associated with an embedding vector and a cosine similarity is used to calculate a semantic relevance score. Sabater et al. [24] presented a one-shot action recognition approach based on a Temporal Convolutional Network (TCN). After normalization of the skeleton stream, they calculate pose features and use the TCN for the generation of motion descriptors. The descriptors at the last frame, assumed to contain all relevant motion from the skeleton-sequence, are used to calculate the distances to the reference samples. Action classes are associated by thresholding the distances. Previous work on multi-modal one-shot action recognition [18] proposed to formulate the one-shot action recognition problem as a deep metric learning problem. Signals originating from various sensors are transformed into images and an encoder is trained using triplet-loss. The focus in that work was on showing the multi-modal applicability, whereas in this work we concentrate on skeleton-based one-shot action recognition.

3. Approach

We propose a novel, compact image representation for skeleton sequences. Additionally, we present an encoder model that learns to project said representations into a metric embedding space that encodes action similarity.

3.1. Problem Formulation

A standard approach for action recognition is trained on a set of classes C, where the training and test sets share the same C classes. Thus, a test set $\mathcal{T}$ share the same classes as the training set $\mathcal{D}$. In a one-shot action recognition setting, C classes are known in an auxiliary training set $\mathcal{D}$, while the evaluation set $\mathcal{T}$ contains U novel classes, providing a single reference sample per class in a reference set $\mathcal{R}$, where $|\mathcal{R}| = U$. We consider the one-shot action recognition problem as a metric learning problem. Our goal is to train a feature embedding $\vec{x} = f_{\Theta}(I)$ with parameters Θ which projects input images $I \in \{0, \dots, 255\}^{H \times W \times 3}$, into a feature representation $\vec{x} \in \mathbb{X}^d$. H denotes the height of the image, W denotes the width of the image in an RGB channel image and d is the given target embedding vector size. The feature representation reflects minimal distances in embedding space for *similar* classes. For defining the similarity we follow [29], where the similarity of two samples $(I_i, \vec{x}_i)$ and $(I_j, \vec{x}_j)$ is defined as $D_{ij} :=< \vec{x}_i, \vec{x}_j >$, where $< \cdot, \cdot >$ denotes the dot product, resulting in an $K \times K$ similarity matrix D.

3.2. Skeleton-DML Representation

We encode skeleton sequences into an image representation. Fig. 2 shows the skeleton as contained in the NTU RGB+D 120 dataset. On a robotic system, these skeletons can be either directly extracted from the RGB-D camera [33] or from a camera image stream using a human-pose estimation approach [2]. The input in our case is a skeleton sequence matrix $S \in \mathbb{R}^{N \times M \times 3}$ where each row vector represents a discrete joint sequence (for N joints) and each column vector represents a sample of all joint positions at one specific time step of a sequence length M. The matrix is transformed to an RGB image $I \in \{0, \dots, 255\}^{H \times W \times 3}$. Note, in contrast to [18, 3] the joint space is not projected to the color channels but unfolded per axis separately like depicted in Fig. 3, and Fig. 4. This results in a dataset $\mathcal{D} = \{(I_i, y_i)\}_{i=1}^{K}$ of K auxiliary training images with one image per skeleton sequence $I_{1,\dots,K}$ with labels $y_i \in \{1, \dots, C\}$. In contrast to the representations used for multimodal action recognition [17] or skeleton based action recognition [28, 16] the proposed representation is more compact. In comparison to [3, 18] our representation separates the joint values for all axes as blocks over the width, keeping all joint values grouped locally together per axis. In [18] the color channels are used to unfold the joint values. As the skeleton-sequence is represented as an image, the model needs to be applied only to a single image for inference.

3.3. Feature Extraction

For better comparability between the approaches we use the same feature extraction method as previously proposed in *SL-DML* [18]. Using a Resnet18 [5] architecture allows us to train a model that converges fast and serves as a good feature extractor for the embedder. The low amount of parameters allows practical use for inference on autonomous mobile robots. Weights are initialized with a pre-trained

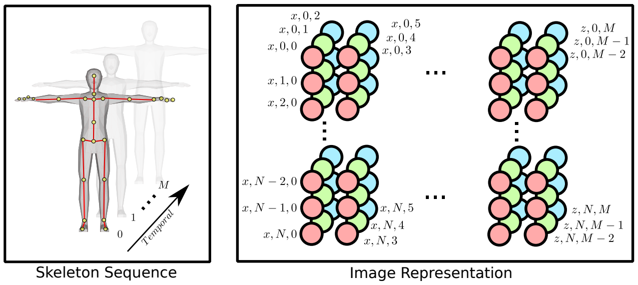

Figure 3. Skeleton-DML skeleton representation. x and z denote the skeleton joint component in joint space, the number of joints is reflected by N, which relates to the height of the image H, the sequence length M relates with the width of the image W. Note, instead of projecting the temporal information throughout the width of the image, we project the joint space locally for each dimension and assemble the joint axis blocks over the width.

model and are optimized throughout the training of the embedder. After the last feature layer we use a two-layer perceptron to transform the features to the given embedding size. The embedder is refined by the metric learning approach.

3.4. Metric Learning

Metric learning aims to learn a function to project an image into an embedding space, where the embedding vectors of similar samples are encouraged to be closer, while dissimilar ones are pushed apart from each other [29]. We use a *Multi-Similarity-Loss* in combination with a *Multi-Similarity-Miner* [29] for mining good pair candidates during training. Positive and negative pairs (by class label) that are assumed to be difficult to push apart in the embedding space are mined. Fig. 5 gives an artificial example of how positive and negative pairs are mined. Positive pairs are constructed by an anchor and positive image pair $\{I_\circ, I_\uparrow\}$ and its embedding $f(I_\circ)$, preferring pairs with a low similarity in embedding space (high distance in embedding space) with the following condition:

$$D^+_{\circ\uparrow} < \max_{k \neq \circ} D_{\circ k} + \epsilon. \tag{1}$$

Similar, if $\{I_\circ, I_\downarrow\}$ is a negative pair, the condition is:

$$D^-_{\circ\downarrow} > \min_{k = \circ} D_{\circ k} - \epsilon, \tag{2}$$

where k is a class label index and ϵ is a given margin.

Note, these conditions support the mining of hard pairs, i.e. a positive pair where the sample still has a high distance in embedding space and a negative pair that still has a low distance in embedding space. This forces sampling that concentrates on the hard pairs. A set of positive images to an anchor image $I_\circ$ are denoted $\mathcal{P}_i$, analog, a set of negative images to $I_\circ$ are denoted $\mathcal{N}_i$.

Given mined positive- and negative pairs allows us integration into the *Multi-Similarity* loss, as derivated by Wang et al. [29]:

$$\begin{aligned} \mathcal{L}_{MS} = \frac{1}{K} \sum_{i=1}^{K} \Bigg\{ & \frac{1}{\alpha} \log \Big[1 + \sum_{k \in \mathcal{P}_i} e^{-\alpha(D_{ik} - \lambda)} \Big] \\ & + \frac{1}{\beta} \log \Big[1 + \sum_{k \in \mathcal{N}_i} e^{\beta(D_{ik} - \lambda)} \Big] \Bigg\}, \end{aligned} \tag{3}$$

where α, β and λ are fixed hyper-parameters.

In contrast to *SL-DML* we do not apply weighting to the classifier- and embedder loss, as no significant improvement has been achieved according to [18]. After the model optimization, associating an action class to a query sample and set of reference samples is now reduced to a nearest-neighbor search in the embedding space. The classifier and encoder are jointly optimized.

3.5. Implementation

Our implementation is based on PyTorch [20], [22]. We tried to avoid many of the metric learning flaws as pointed out by Musgrave et al. [19] by using their training setup and hyperparameters, where applicable. Key differences are that we use a Resnet18 [5] architecture and avoid the proposed four-fold cross validation for hyperparameter search in favour of better comparability to the proposed one-shot protocol on the *NTU RGB+D 120* dataset [12]. Note, we did not perform any optimization of the hyperparameters. A batch size of 32 was used on a single Nvidia GeForce RTX 2080 TI with 11GB GDDR-6 memory. We trained for 100 epochs with initialized weights of a pre-trained Resnet18 [5]. For the multi similarity miner we used an epsilon of 0.05 and a margin of 0.1 for the triplet margin loss. A RMSProp optimizer with a learning rate of 10^{-6} was used in all optimizers. The embedding model outputs a 128 dimensional embedding.

4. Experiments

We used skeleton sequences from the *NTU RGB+D 120* [12] dataset for large scale one-shot action recognition.

The dataset is split into an auxiliary training set, representing action classes that are used for training, and an evaluation set with the classes used for testing. In the one-shot protocol, the evaluation set does only contain novel, previously unseen, actions. One sample of each evaluation class serves as reference demonstration. This protocol is based on the one proposed by [12] for the *NTU RGB+D 120* dataset. First, we trained a model on the auxiliary training set. The resulting model transforms skeleton-sequences encoded as an image representation into embeddings for the reference samples and evaluation samples. We then calculate the nearest neighbor from the evaluation set embeddings to the

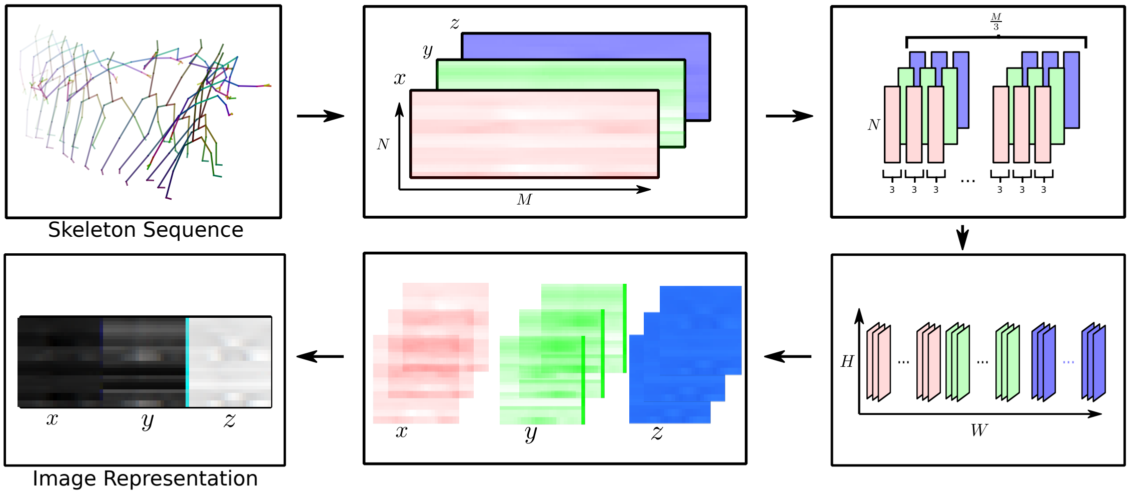

Figure 4. Exemplary representation for a throwing activity of the *NTU-RGB+D 120* dataset. A skeleton-sequence serves an input and can be represented as an image directly [3, 17]. Our Skeleton-DML representation groups x-, y-, z joint values locally in $\frac{M}{3}$ blocks per axis and assembles them into the final image representation. All axis blocks are laid out aside.

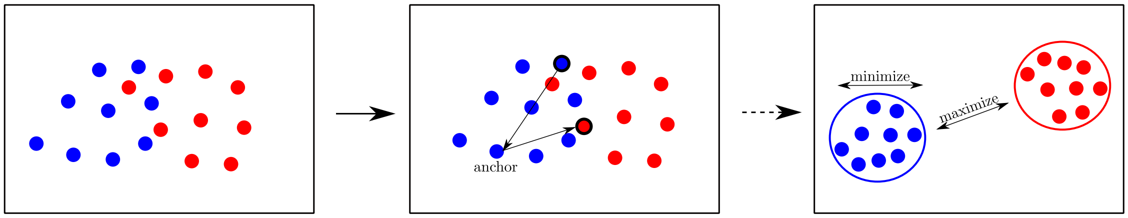

Figure 5. A possible intermediate state of the embeddings during the training process of two classes (left). During training, pairs, that are difficult to push apart in embedding space, are mined (middle). Given the blue anchor sample, the most difficult positive pair is the blue sample with the highest distance in embedding space. Similar, the closest red sample in embedding space is the corresponding negative sample. The overall goal is to separate the samples in embedding space (right) by minimizing the inter-class scatter and maximize the intra-class distance to the class centers in embedding space.

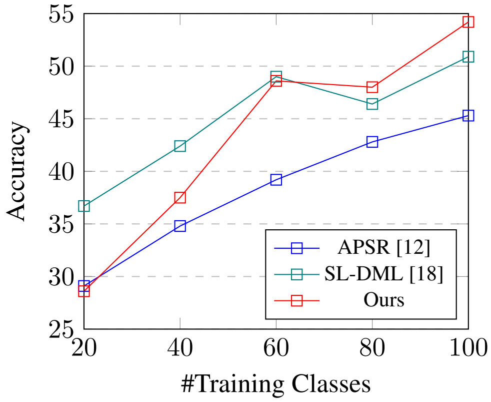

Figure 6. Result graph for increasing auxiliary training set sizes.

reference embeddings. As the embeddings encode action similarity, we can estimate to which reference sample the given test sample comes closest. Beside the standard one-shot action protocol and experiments with dataset reduction, we give an ablation study that gives a hint on which combination of embedding size, loss, transformation and representation are yielding best results with our approach. Further, we integrated various related skeleton-based image representations that have been previously proposed for action recognition into our one-shot action recognition approach to compare them.

4.1. Dataset

The *NTU RGB+D 120* [12] dataset is a large scale action recognition dataset containing RGB-D image streams and skeleton estimates. The dataset consists of 114,480 sequences containing 120 action classes from 106 subjects in 155 different views. We follow the one-shot protocol as described by the dataset authors. The dataset is split into two parts: an auxiliary training set and an evaluation set. The action classes of the two parts are distinct. 100 classes are used for training that define the auxiliary set, 20 classes are used for testing that define the evaluation set. A single sample per class from the evaluation set serves as reference sample. The unseen classes and reference samples are doc-

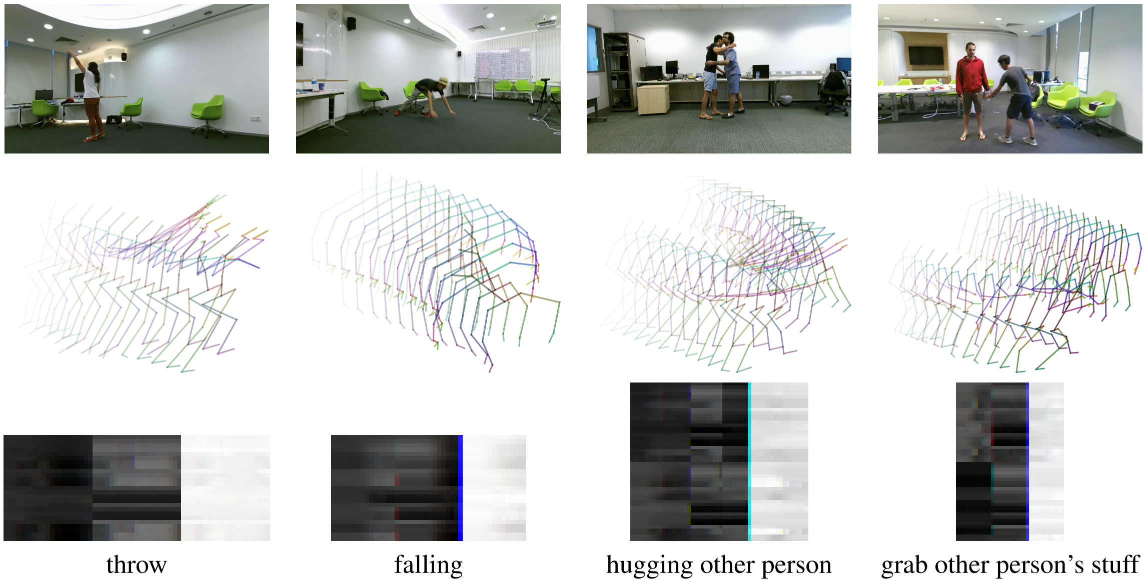
throw falling hugging other person grab other person’s stuff

Figure 7. From top to bottom: A RGB Frame, the corresponding skeleton sequences and the image representation of those sequences are shown. The latter is used in our one-shot action recognition approach. The first two sequences contain single person activities, whereas the remaining two contain two person interactions. The *grab other person’s stuff* sequence was shorter than the *hugging other person* sequence.

Table 1. One-shot action recognition results on the *NTU RGB+D 120* dataset.

Approach	Accuracy [%]
Attention Network [15]	41.0
Fully Connected [15]	42.1
Average Pooling [14]	42.9
APSR [12]	45.3
TCN [24]	46.5
SL-DML [18]	50.9
Ours	**54.2**

Table 2. Results for different auxiliary training set sizes for one-shot recognition on the *NTU RGB+D 120* dataset in %.

#Train Classes	APSR [12]	*SL-DML* [18]	Ours
20	29.1	**36.7**	28.6
40	34.8	**42.4**	37.5
60	39.2	**49.0**	48.6
80	42.8	46.4	**48.0**
100	45.3	50.9	**54.2**

umented in the accompanied dataset repository[1]. *A1, A7, A13, A19, A25, A31, A37, A43, A49, A55, A61, A67, A73, A79, A85, A91, A97, A103, A109, A115* are previously unseen. As reference, the demonstration for filenames starting with *S001C003P008R001** are used for actions with IDs below 60 and *S018C003P008R001** for actions with IDs above 60. As no hold-out validation set is defined, we defined a separate validation set separated from the auxiliary training set by using the following classes during development for validation: *A2, A8, A14, A20, A26, A32, A38, A44, A50, A56, A62, A68, A74, A80, A86, A92, A98, A104, A110, A116*. One-shot action recognition results are given in Table 1. Like Liu et al. [12] we also experimented with the effect of the auxiliary training set reduction. Results are given in Fig. 6 and Table 2. In addition, we analyze different representations in Table 4 and the influence of different embedding vector sizes, metric losses and augmentations on two representations more detailed in Table 3.

4.2. Training Set Size Reduction

An interesting question that comes up when evaluating one-shot action recognition approaches is how much auxiliary training classes are required to get a certain performance. Liu et al. [12] already proposed to evaluate the one-shot action recognition approach with varying auxiliary training set sizes. Aligned with Liu et al. [12] we use auxiliary training sets containing 20, 40, 60, 80 auxiliary training classes while remaining a constant evaluation set size of 20. For practical systems, where only a limited amount of training data is available, this evaluation can give an important insight about which performance can be achieved with lower amounts of provided training data. It is also interesting to observe how an approach performs when adding

[1] https://github.com/shahroudy/NTURGB-D

Table 3. Ablation study for our proposed one-shot action recognition with different representations, embedding sizes, losses and augmentations. Results are given for a training over 200 epochs. Units are in %.

Representation	128	256	512	Transform	Loss
SL-DML [18]	55.2	50.6	52.7	None	MS
SL-DML [18]	51.5	51.7	54.0	None	TM
SL-DML [18]	51.8	55.3	55.8	Rot	MS
SL-DML [18]	53.6	54.8	55.5	Rot	TM
Ours	54.7	51.5	53.1	None	MS
Ours	47.5	51.9	54.0	None	TM
Ours	55.3	58.0	**58.6**	Rot	MS
Ours	56.0	55.1	56.1	Rot	TM

more training data. Table 2 and Fig. 6 give results for different training set sizes for *SL-DML* [18], *APSR* [12] and our Skeleton-DML approach, while remaining a static validation set. With just 20 training classes, our approach performs comparably to the *APSR* approach. With a small amount of training classes, the *SL-DML* approach performs best. In our experiments, Skeleton-DML performs better when providing a larger auxiliary training set size. At an auxiliary training set size of 60 classes, our approach performs comparably well to *SL-DML*. With 80 classes in the auxiliary training set our approach starts outperforming *SL-DML*. It is interesting to note that, aligned with the results from *SL-DML*, our approach seems to be confused by the 20 extra classes that are added to the 60 classes.

4.3. Ablation Study

To distill the effects of the components, we report their individual contributions. We examine influence of the representation, augmentation method and different resulting embedding vector sizes. Inspired by Roth et al. [23] we experiment with different embedding vector sizes of 128, 256, 512. In addition, we included the *SL-DML* representation, compare a Triplet Margin loss (TM) and a Multi-Similarity loss (MS) and included an augmentation with random rotations of 5°. In total, 24 models were trained for this ablation study. We trained these models for 200 epochs, as we expected longer convergence due to the additional augmented data. Results are given in Table 3. In the table, we highlight important results. We highlight interesting results by different colors in the table (best result without augmentation (55.2%), embedding size of 128 (56.0%), embedding size of 256 (58.0%), TM loss (56.1%), overall, MS loss, augmentation, embedding size of 512 (58.6%)). For *SL-DML* the augmentation had a positive influence with higher embedding vector sizes of 512. Whereas the augmentation with embedding sizes of 128 only improved with the TM loss. With the MS loss and a low embedding size the augmentation did lower the result. For our Skeleton-DML representation the augmentation improved the results throughout the experiments for both losses. The best results without augmentation were achieved by the *SL-DML* representation with an embedding vector of size 128 and a MS loss. The overall best results were achieved with a MS loss and embedding vector size of 512 and augmentation by rotation using the Skeleton-DML representation, which improved the results of 4.4% over our approach under a comparable training setup as *SL-DML*.

Table 4. Ablation study for different representations.

Representation	Accuracy [%]
Skepxel [11]	29.6
SkeleMotion Orientation [1]	34.4
SkeleMotion MagnitudeOrientation [1]	39.2
TSSI [32]	41.0
Gimme Signals [17]	41.5
SkeleMotion Magnitude [1]	44.4
SL-DML [18]	50.9
Ours	**54.2**

4.4. Comparison with Related Representations

To support the effectiveness of our proposed representation in a metric learning setting we compare against other skeleton-based image representations. We use the publicly available implementation for the *SkeleMotion* [1], *SL-DML* [18], Gimme Signals [17] and re-implementations of the *TSSI* [32] and *Skepxels* [11] representations to integrate them into our metric learning approach. These representations have been described in Section 2 more detailed.

The overall training procedure was identical as all models were trained with the parameters described in Section 3.5. The experiment only differed in the underlying representation. Results for the representation comparison are given in Table 4. While most of the representations initially target action recognition and are not optimized for one-shot action recognition, they are still good candidates for integration in our metric learning approach. We did not re-implement the individual architecture proposed by the different representations but decided to use the Resnet18 architecture for better comparability.

Our Skeleton-DML approach shows best performance followed by *SL-DML*. The *SkeleMotion* Magnitude [1] representation transfers well from an action recognition setting to a one-shot action recognition setting. Interesting to note is that the *SkeleMotion* Orientation [1] representation, while achieving comparable results in the standard action recognition protocol, performs 10% worse than the same representation encoding the magnitude of the skeleton joints. An early fusion of magnitude and orientation on a representa-

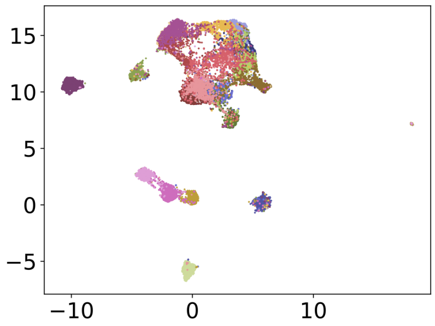

Figure 8. *UMAP* embedding visualization for our approach. Classes are: drink water ●, throw ●, tear up paper ●, take off glasses ●, reach into pocket ●, pointing to something with finger ●, wipe face ●, falling ●, feeling warm ●, hugging other person ●, put on headphone ●, hush (quite) ●, staple book ●, sniff (smell) ●, apply cream on face ●, open a box ●, arm circles ●, yawn ●, grab other person's stuff ●, take a photo of other person ●.

tion level did not improve the Skelemotion representation, but yields a result in between both representations. Similar observations have been made in [18] by the fusion of inertial and skeleton sequences. The lower performing modality adds uncertainty to the resulting model in our one-shot setting.

A *UMAP* embedding of all evaluation samples is shown in Fig. 8 for our Skeleton-DML approach. Our approach shows better capabilities in distinguishing the actions throw and arm circles. In our approach those clusters can be separated quite well whereas *SL-DML* struggles to discriminate those two classes.

4.5. Result Discussion

We evaluated our approach in an extensive experiment setup. Aside from lower performance on lower amounts of classes for training, our approach outperformed other approaches. For fair comparison we report the result of 3.3% over *SL-DML* for training with 100 epochs and without augmentation, as under these conditions the *SL-DML* result was reported. With augmentation and training for 200 epochs, we could improve the baseline for 7.7%. Our approach learns an embedding model that captures semantic relevance from joint movements well. E.g. Skeleton-DML differentiates well between activities that primarily contain hand- or leg-movements. Interactions between multiple person and single person activities are also separated well. Activities to which similar joint movements contribute to are still challenging. These are the activities that are formed by the main cluster in Fig. 8.

5. Conclusion

We presented a one-shot action recognition approach based on the transformation of skeleton sequences into an image representation. On the image representations, an embedder is trained which projects the images into an embedding vector. Distances between encoded actions reflect semantic similarities. Actions can then be classified, given a single reference sample, by finding the nearest neighbour in embedding space. In an extensive experiment setup we compared different representations, losses, embedding vector sizes and augmentations. Our representation remains flexible and yields improved results over *SL-DML*. Additional augmentation by random 5 degree rotations have shown to further improve the results. We found the overall approach of transforming skeleton sequences into image representations for one-shot action recognition by metric learning a promising idea that allows future research into various directions like finding additional representations, augmentation methods or mining and loss approaches. Especially in robot applications one-shot action recognition approaches have the potential to improve human robot interaction by allowing robots to adapt to unknown situations. The required computational cost for our approach is low, as only a single image representations of the skeleton-sequence needs be embedded by a comparably slim Resnet18-based embedder.

References

[1] Carlos Caetano, Jessica Sena de Souza, François Brémond, Jefersson A. dos Santos, and William Robson Schwartz. Skelemotion: A new representation of skeleton joint sequences based on motion information for 3d action recognition. In *16th IEEE International Conference on Advanced Video and Signal Based Surveillance, AVSS 2019, Taipei, Taiwan, September 18-21, 2019*, pages 1–8. IEEE, 2019.

[2] Zhe Cao, Gines Hidalgo, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Openpose: realtime multi-person 2d pose estimation using part affinity fields. *IEEE transactions on pattern analysis and machine intelligence*, 43(1):172–186, 2019.

[3] Yong Du, Yun Fu, and Liang Wang. Skeleton based action recognition with convolutional neural network. In *2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR)*, pages 579–583. IEEE, 2015.

[4] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML*, 2017.

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[6] Shawn Hershey, Sourish Chaudhuri, Daniel PW Ellis, Jort F Gemmeke, Aren Jansen, R Channing Moore, Manoj Plakal, Devin Platt, Rif A Saurous, Bryan Seybold, et al. Cnn architectures for large-scale audio classification. In *2017 ieee international conference on acoustics, speech and signal processing (icassp)*, pages 131–135. IEEE, 2017.

[7] Elad Hoffer and Nir Ailon. Deep metric learning using triplet network. In *International Workshop on Similarity-Based Pattern Recognition*, pages 84–92. Springer, 2015.

[8] Jian-Fang Hu, Wei-Shi Zheng, Lianyang Ma, Gang Wang, Jianhuang Lai, and Jianguo Zhang. Early action prediction by soft regression. *IEEE transactions on pattern analysis and machine intelligence*, 41(11):2568–2583, 2018.

[9] Tae Soo Kim and Austin Reiter. Interpretable 3d human action analysis with temporal convolutional networks. In *2017 IEEE conference on computer vision and pattern recognition workshops (CVPRW)*, pages 1623–1631. IEEE, 2017.

[10] Chao Li, Qiaoyong Zhong, Di Xie, and Shiliang Pu. Skeleton-based action recognition with convolutional neural networks. In *2017 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*, pages 597–600. IEEE, 2017.

[11] Jian Liu, Naveed Akhtar, and Ajmal Mian. Skepxels: Spatio-temporal image representation of human skeleton joints for action recognition. In *CVPR Workshops*, 2019.

[12] Jun Liu, Amir Shahroudy, Mauricio Lisboa Perez, Gang Wang, Ling-Yu Duan, and Alex Kot Chichung. Ntu rgb+ d 120: A large-scale benchmark for 3d human activity understanding. *IEEE transactions on pattern analysis and machine intelligence*, 2019.

[13] Jun Liu, Amir Shahroudy, Dong Xu, Alex C Kot, and Gang Wang. Skeleton-based action recognition using spatio-temporal lstm network with trust gates. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):3007–3021, 2017.

[14] Jun Liu, Gang Wang, Ling-Yu Duan, Kamila Abdiyeva, and Alex C Kot. Skeleton-based human action recognition with global context-aware attention lstm networks. *IEEE Transactions on Image Processing*, 27(4):1586–1599, 2017.

[15] Jun Liu, Gang Wang, Ping Hu, Ling-Yu Duan, and Alex C Kot. Global context-aware attention lstm networks for 3d action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1647–1656, 2017.

[16] Mengyuan Liu, Hong Liu, and Chen Chen. Enhanced skeleton visualization for view invariant human action recognition. *Pattern Recognition*, 68:346–362, 2017.

[17] Raphael Memmesheimer, Nick Theisen, and Dietrich Paulus. Gimme signals: Discriminative signal encoding for multimodal activity recognition. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Las Vegas, NV, USA, 2020. IEEE.

[18] Raphael Memmesheimer, Nick Theisen, and Dietrich Paulus. Signal level deep metric learning for multimodal one-shot action recognition. *arXiv preprint arXiv:2004.11085*, 2020.

[19] Kevin Musgrave, Serge Belongie, and Ser-Nam Lim. A metric learning reality check. In *European Conference on Computer Vision*, pages 681–699. Springer, 2020.

[20] Kevin Musgrave, Ser-Nam Lim, and Serge Belongie. Pytorch metric learning. `https://github.com/KevinMusgrave/pytorch-metric-learning`, 2019.

[21] Jordi Pages, Luca Marchionni, and Francesco Ferro. Tiago: the modular robot that adapts to different research needs. In *International workshop on robot modularity, IROS*, 2016.

[22] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, pages 8024–8035, 2019.

[23] Karsten Roth, Timo Milbich, Samarth Sinha, Prateek Gupta, Bjorn Ommer, and Joseph Paul Cohen. Revisiting training strategies and generalization performance in deep metric learning. In *International Conference on Machine Learning*, pages 8242–8252. PMLR, 2020.

[24] Alberto Sabater, Laura Santos, Jose Santos-Victor, Alexandre Bernardino, Luis Montesano, and Ana C Murillo. One-shot action recognition towards novel assistive therapies. *arXiv preprint arXiv:2102.08997*, 2021.

[25] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823, 2015.

[26] Amir Shahroudy, Jun Liu, Tian-Tsong Ng, and Gang 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.

[27] Jiang Wang, Yang Song, Thomas Leung, Chuck Rosenberg, Jingbin Wang, James Philbin, Bo Chen, and Ying Wu. Learning fine-grained image similarity with deep ranking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1386–1393, 2014.

[28] Pichao Wang, Wanqing Li, Chuankun Li, and Yonghong Hou. Action recognition based on joint trajectory maps with convolutional neural networks. *Knowledge-Based Systems*, 158:43–53, 2018.

[29] Xun Wang, Xintong Han, Weilin Huang, Dengke Dong, and Matthew R Scott. Multi-similarity loss with general pair weighting for deep metric learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5022–5030, 2019.

[30] Nicolai Wojke and Alex Bewley. Deep cosine metric learning for person re-identification. In *2018 IEEE winter conference on applications of computer vision (WACV)*, pages 748–756. IEEE, 2018.

[31] Takashi Yamamoto, Tamaki Nishino, Hideki Kajima, Mitsunori Ohta, and Koichi Ikeda. Human support robot (hsr). In *ACM SIGGRAPH 2018 emerging technologies*, pages 1–2. 2018.

[32] Zhengyuan Yang, Yuncheng Li, Jianchao Yang, and Jiebo Luo. Action recognition with spatio–temporal visual attention on skeleton image sequences. *IEEE Transactions on Circuits and Systems for Video Technology*, 29(8):2405–2415, 2018.

[33] Zhengyou Zhang. Microsoft kinect sensor and its effect. *IEEE Multim.*, 19(2):4–10, 2012.