A Comparison of Few-shot Classification of Human Movement Trajectories

Lisa Gutzeit
Robotics Research Group, University of Bremen, Bremen, Germany

Keywords: Few-shot Learning, Movement Recognition, Human Movement Analysis, k-Nearest Neighbor, Long Short-term Memory, Hidden Markov Model.

Abstract: In the active research area of human action recognition, a lot of different approaches to classify behavior have been proposed and evaluated. However, evaluations on movement recognition with a limited number of training examples, also known as Few-shot classification, are rare. In many applications, the generation of labeled training data is expensive. Manual efforts can be reduced if algorithms are used which give reliable results on small datasets. In this paper, three recognition methods are compared on gesture and stick-throwing movements of different complexity performed individually without detailed instructions in experiments in which the number of the examples used for training is limited. Movements were recorded with marker-based motion capture systems. Three classification algorithms, the Hidden Markov Model, Long Short-Term Memory network and k-Nearest Neighbor, are compared on their performance in recognition of these arm movements. The methods are evaluated regarding accuracy with limited training data, computation time and generalization to different subjects. The best results regarding training with a small number of examples and generalization are achieved with LSTM classification. The shortest calculation times are observed with k-NN classification, which shows also very good classification accuracies on data of low complexity.

1 INTRODUCTION

Classification of human movements is of high interest in many applications. For example in man machine interaction, human behaviors, intentions and habits have to be better understood to facilitate future approaches in which humans closely collaborate with robotic systems. To make an intuitive interaction possible, methods are needed which analyze naturally performed human behavior.

In the last decades, many approaches to analyze video or image data to understand human behaviors have been presented (Poppe, 2010). Most of these approaches benefit from a huge amount of available data. In contrast to human activity recognition in the wild, there are applications in which smaller movement entities, such as a specific type of grasping, need to be detected. For example these movement entities can be used in robotic applications to transfer basic movement types to a robotic system using, e.g., learning from demonstration (LfD) (see, e.g., (Argall et al., 2009) for an LfD overview). To acquire training data for these applications, movement demonstrations need to be recorded, pre-processed and manually labeled. These efforts can be minimized if so-called Few-shot classification methods are used, i.e. methods that can recognize various behaviors with a small number of training examples. Additionally, by using such methods training time as well as the resources needed for re-training, which can be used, e.g., to address newly observed movements, are minimized.

In (Gutzeit et al., 2019b), small entities of human manipulation movements haven been detected at high accuracy in different behavior demonstrations with ≤ 10 examples per class in the training data. For this, recorded movements were automatically segmented into manipulation building blocks characterized by a bell-shaped velocity profile of the hand, see (Senger et al., 2014) for details. For example, a ball-throwing movement was segmented into its three building blocks, *strike out*, *throw*, and *swing out*. To recognize these building blocks, a classification accuracy of 80% could be achieved with a simple 1-Nearest Neighbor classifier with only 4 training examples per class (Gutzeit et al., 2019b). Using this approach, detected movements in pick-and-place, lever-pulling and different throwing tasks have been successfully transfered to various robotic sys-

Gutzeit, L.
A Comparison of Few-shot Classification of Human Movement Trajectories.
DOI: 10.5220/0010244002430250
In *Proceedings of the 10th International Conference on Pattern Recognition Applications and Methods (ICPRAM 2021)*, pages 243-250
ISBN: 978-989-758-486-2

tems (Gutzeit et al., 2018; Gutzeit et al., 2019a).

In this paper, the recognition of different human arm movements using Few-shot classification is investigated more closely. Three different algorithms which are widely used for human action recognition, the k-Nearest Neighbor classifier, classification based on Hidden Markov Models and Long Short-Term Memory networks, are compared. Furthermore, the generalization of these classifiers to the movements of persons whose demonstrations were not part of the training data is analyzed. For evaluation, two different datasets, containing different gestures and the building blocks of a throwing movement respectively, are used. For data recording we use marker-based motion tracking systems, which measure the positions of important points on the body directly at a high precision.

This paper is organized as follows: In section 2, an overview about related work is given. In section 3, the features and methods used for classification are described as well as the evaluation approach. The data recorded and the evaluation results are presented in section 4 and section 5 respectively. At the end, the results are discussed and a conclusion is given.

2 RELATED WORK

Human action recognition is an active research area with a lot of different applications and methods. Most approaches are based on the analysis of video or RGB-D data in applications such as the detection of tackles in soccer games, support of elderly in their homes, or gesture recognition in video games (Poppe, 2010). In these approaches large efforts have to be put into the detection of the human and its posture in the measured data streams. Afterwards, the observed actions are classified with algorithms such as Support Vector Machines, or their probabilistic variant the Relevance Vector Machines, Hidden Markov Models (HMMs) or k-Nearest Neighbors (k-NN), see (Poppe, 2010) for a detailed overview.

In the last decades, HMMs were widely used to classify human actions and gestures. For example in (Stefanov et al., 2010) and (Aarno and Kragic, 2008), HMMs were used to recognize human intentions in teleoperation scenarios. Borghi et al. propose an online double-stage Multiple Stream Discrete HMM to classify gestures from 3D joint positions acquired with a Kinect (Borghi et al., 2016). With this approach, high classification accuracies could be achieved on three public and a new recorded data set containing different actions created for human computer interaction.

Recently, neural network based approaches became popular in all pattern recognition domains. Patsdu et al. compared a neural network with a Support Vector Machine, a decision tree, and Naive Bayes to distinguish the movement patterns *stand*, *sit down*, and *lie down* recorded with a Kinect camera (Patsadu et al., 2012). In the huge data set with more than 10.000 recordings, the best performance was reached with the neural network approach. Long-term motions in video sequencs were detected in (Shi et al., 2017) using a method based on a CNN-RNN network. To handle un-reliable data, Liu et al. introduced a new gating algorithm for Long-Short Term Memories (LSTMs) (Liu et al., 2017). Spatial and temporal dependencies between joints are learned to recognize human action in skeleton data. Unreliable data, which can result from noisy data or occlusions, are handled with a newly introduced trust gate added to the LSTM.

However, the majority of the approaches in the literature are applied to precisely specified movements. The performance with respect to naturally and intuitively performed movements is not analyzed. Furthermore, many approaches rely on huge sets of labeled data. If these are not available for a certain application, the training datasets have to be manually generated, which requires a large human effort. To reduce this effort, algorithms which give reliable results on small dataset sizes are beneficial. This new research area is known as Few-shot learning, a survey is presented in (Wang et al., 2020).

3 METHODS

In this section, the features of the movement trajectories used to distinguish different motions are described as well as the classification approaches with their parameter configurations compared in this paper.

3.1 Feature Extraction

In this work, the human movement is recorded with markers placed on hand, elbow, and shoulder of the subject. The positions of the markers can be seen in Fig. 2 and Fig. 3. All marker positions are transformed into a coordinate system on the back of the subject to make the positions independent from the position of the subject in the global coordinate frame. From each marker, the 3D position and the absolute velocity are used as features. Depending on the tracking system, these values are directly measured or can be calculated easily from the raw data. Additional features are the orientation of the hand and the angle between lower and upper arm (*elbow joint*) and the angle between upper arm and the line connecting the shoulder and the marker on the back (*shoulder joint*) with their corresponding velocities. All feature trajectories are interpolated to a length of 25 using Spline interpolation. Since the range of the individual features varies, all features are normalized to values in the range $[0,1]$.

3.2 Classification Methods

3.2.1 k-Nearest Neighbor

We use k-NN for comparison in this work, because it showed very good results in classification of small movement units on small training dataset sizes (Gutzeit et al., 2019b; Gutzeit et al., 2019a). Additionally, the algorithm does not need much parameter tuning, as it has just one hyper-parameter k. To classify the recorded data sequences with k-NN, the feature trajectories for each movement recording are transformed into a single feature vector. The closest neighbor of each data sample is determined using Euclidean distance.

3.2.2 Hidden Markov Model

HMMs are very common probabilistic models for time series data. A detailed introduction is, e.g., given by Bishop (Bishop, 2006). In this paper, one HMM with Gaussian emissions is trained for each class in the data using the Baum-Welch algorithm. A new data sample is assigned to the label of the HMM from which it is most likely generated. For each HMM, the number of hidden states h has to be set.

3.2.3 Long Short-term Memory

LSTMs are artificial recurrent neural networks especially designed to process time series data, firstly presented in (Hochreiter and Schmidhuber, 1997). In this paper, we use a simple structure with one LSTM layer. The input layer contains one neuron for each feature, which is fully connected to the LSTM layer. As output, a Dense layer with softmax activation function is used, which has a single neuron for each class. During training, the categorical cross entropy is used as error function. To prevent over-fitting, we apply early stopping and stop training if the accuracy on a validation dataset did not increase in the last p epochs, where p is called patience value. For this architecture, we compare different numbers of cells, c, different batch sizes b and patience values p.

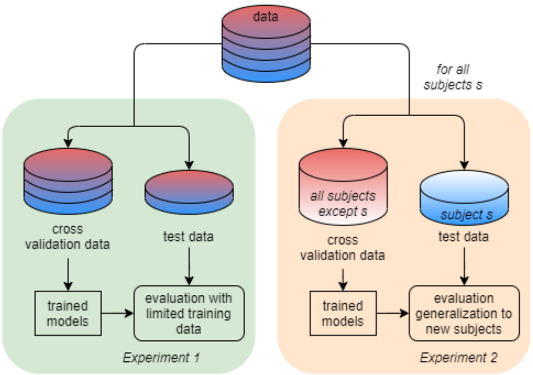

Figure 1: Schematic overview of the evaluation approach described in section 3.3.

3.3 Evaluation Approach

The three algorithms described in section 3.2 are compared with respect to classification with a small number of training examples, computation times and generalization to different subjects on data of different complexity. For this, two experiments were designed. A schematic overview is given in Fig. 1.

In experiment 1 the classification accuracy on small training sizes is evaluated. For this, i samples of each class are randomly selected and used to train the classifier. The remaining samples are used for testing. This is repeated 10 times for each $i \in 1,....,10,15,20$. The final models are tested on a test set which was not part of the cross-validation data, consisting of 10% of the original dataset. The cross-validation is done for each classifier with different hyper-parameter values.

The generalization to new subjects is evaluated in experiment 2. For this, the classifiers are validated on the data of all subjects except one, using a limited training set containing i randomly selected examples of each class for each of the remaining subjects. In the validation data, the number of examples per class is fixed to 10 to avoid unbalanced classes. Final models are tested on the samples of the excluded subject which movements were left out for training. This is repeated 10 times for each subject and each limit i.

4 EXPERIMENTAL DATA

4.1 Gesture Data

For the first analysis, different gestures were recorded with the Xsens MVN Awinda[1] sensor suit, which measures angular velocities and accelerations, from

[1] For more details refer to the vendors websites: https://www.xsens.com and https://www.qualisys.com

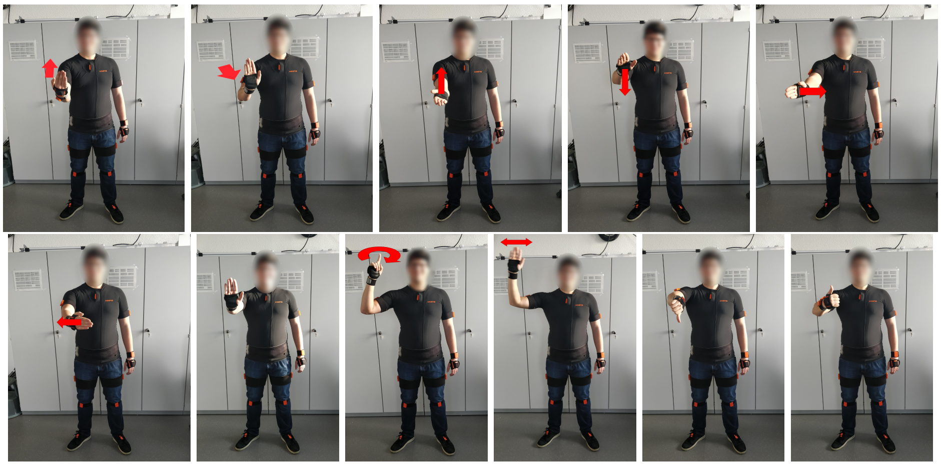

Figure 2: Recorded gestures. Arrows indicate the direction of the movement. The performed gesture from top right to bottom left are: *come closer*, *move backwards*, *move upwards*, *move downwards*, *move left*, *move right*, *stop*, *rally*, *hello*, *thumbs up*, and *thumbs down*.

which positions and velocities can be calculated, with inertial measurement units at 60 Hz. 11 gestures were recorded from 6 subjects. The gestures are shown in Fig. 2. The dataset consists of simple gestures such as *stop* or *thumbs up* and of more complex gestures with repetitive movements like *rally*. Each gesture was demonstrated one time to the subjects before recording. Afterwards, each subject performed each gesture with the instruction to move naturally. For recurring gestures, such as *rally*, the number of repetitions was not specified but intuitively selected by the subject.

In total, each subject performed each gesture 10 – 11 times. For one subject between 30 and 50 repetitions of each gesture were recorded. The gesture trajectories have a length between 17 and 188 time points. In total, 1045 examples of different gesture executions were available for evaluation.

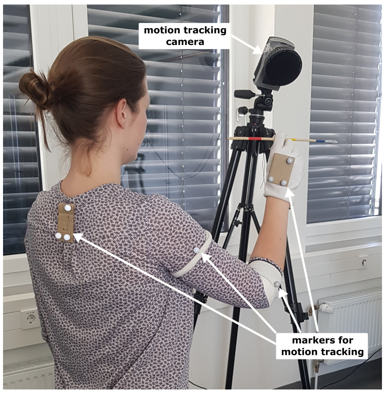

Figure 3: Stick-throwing setup. Positions of markers attached on the arm and the back of the subject are recorded using a camera based motion tracking system (Image taken from (Gutzeit et al., 2019a) with permission).

4.2 Stick-throwing Data

As a second dataset we chose throwing demonstrations, previously used in (Gutzeit et al., 2019a), in which the task was to throw a stick into a box. The movements of 7 subjects were recorded with a Qualisys motion tracking system[1] which uses infrared light reflecting markers. Markers were attached to the right hand, elbow, shoulder and back of the subjects, as shown in Fig. 3. The marker positions were measured with several cameras at 60 Hz. Three markers instead of one were attached to the hand and the back to track also the orientation. The subjects performed between 41 and 246 throws, which result in a total of 697 stick-throwing samples.

The throwing recordings were automatically segmented using a velocity-based probabilistic segmentation presented in (Senger et al., 2014) into basic movement units with a bell-shaped velocity. This resulted in 2913 movement segments. Afterwards, the resulting segments were manually labeled into the movement classes *strike out*, *throw*, *swing out*, and *idle*. Segments which could not be assigned to one of these classes were not considered. This resulted in 2233 labeled segments. The segment trajectories of the main movements have a length between 10 and 136 time points, where segments of the class *idle* have a length between 6 and 269. For each class between 358 and 655 movement examples were available.

4.3 Complexity of the Datasets

In this section, the two datasets are compared with respect to their structure and variety. For this, the

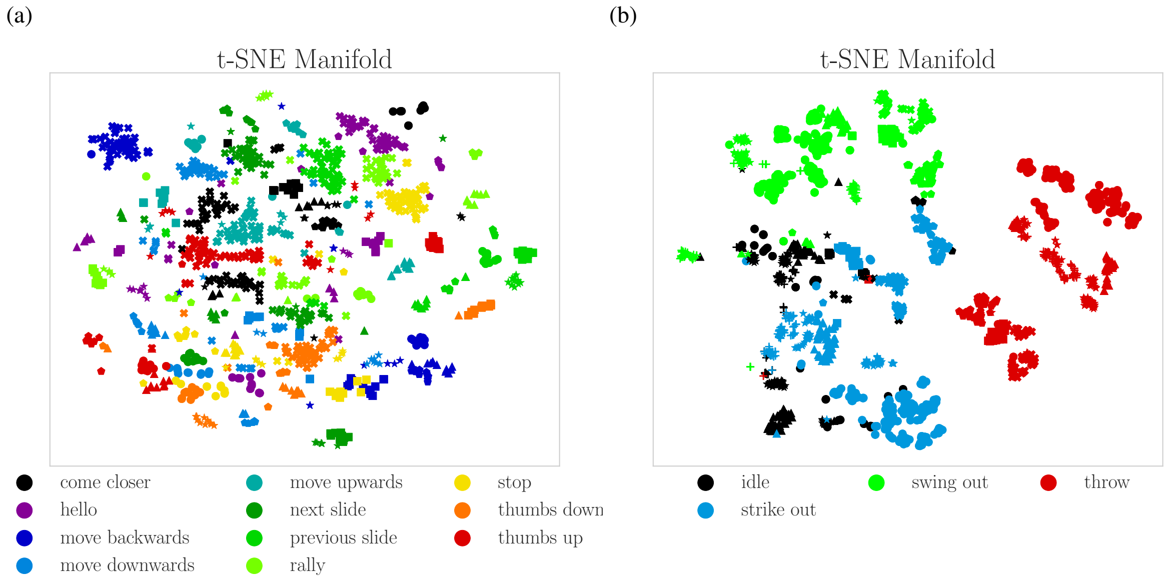

Figure 4: T-SNE manifolds of the gesture data (a) and the stick-throwing data (b). Each movement class can be identified by a different color, samples of the different subjects have different markers.

features of all recordings are transformed into a two dimensional manifold using t-distributed Stochastic Neighbor Embedding (van der Maaten and Hinton, 2008). The result, in which samples with a low feature distance are close, is shown in Fig. 4. The manifold transformation of the gesture data can be seen in Fig. 4a. Although clusters for the different movement classes can be observed, the clusters of the 11 classes clearly overlap. Additionally, movement trajectories of different subjects of the same gesture can be separated in this visualization, as the subject samples show clusters within one gesture class. Thus, the generalization to new subjects is a challenging tasks for this heterogeneous dataset.

On the other hand, the movement classes in stick-throwing data are separated more clearly, see Fig. 4b. Although samples of the same class by different subjects can be distinguished in this data, too, the distances to the other classes are higher. Only the two classes *idle* and *strike out* overlap in the manifold. This shows the much lower complexity of this data compared to the gesture data. This has several reasons. First, the gesture data contains more classes and some of them are very similar in their execution. For example the movement classes *thumbs up* and *thumbs down* differ only in the orientation of the hand. In the stick-throwing dataset the task is to throw a stick to a certain position. This is in contrast to the movements in the gesture set a goal-directed behavior, in which less variations can be assumed. Furthermore, the stick-throwing data is segmented into its main movement blocks characterized by a bell-shaped velocity profile as introduced in (Senger et al., 2014), which further reduces complexity.

5 RESULTS

5.1 Gesture Classification

The validations on the gesture data were performed with hyper-parameters set to $k \in [1,3,5,7,10,15,20]$ for k-NN, $h \in [5,10,15,20,25]$ for HMM, and $c \in [5,10,15,25,30,40,50,70]$ for LSTM with $b \in [8,16,32,128]$ and $p \in [5,10,15]$.

In experiment 1, the classifiers are validated with a limited number of training examples per class. The results with limit 10 are shown in Fig. 5. The hyper-parameters b and p of the LSTM classifier are fixed to $b = 16$ and $p = 10$, which gave the highest accuracies. With a maximum of 10 examples per class in the training data, the best result is achieved with the LSTM classifier with $c = 50$ cells, leading to a mean accuracy of 68%(±0.07%). k-NN with $k = 1$ has a similar mean accuracy (67%(±0.4%)). HMM classification does not achieve an accuracy above 60% with this small training size. k-NN has the fastest calculation times, including short prediction times. The training time of the best LSTM network is around 1000 times slower, but after training the prediction is similar to 1-NN classification. With HMM training and prediction takes even longer. Note the different axis scalings in the visualization of the computation

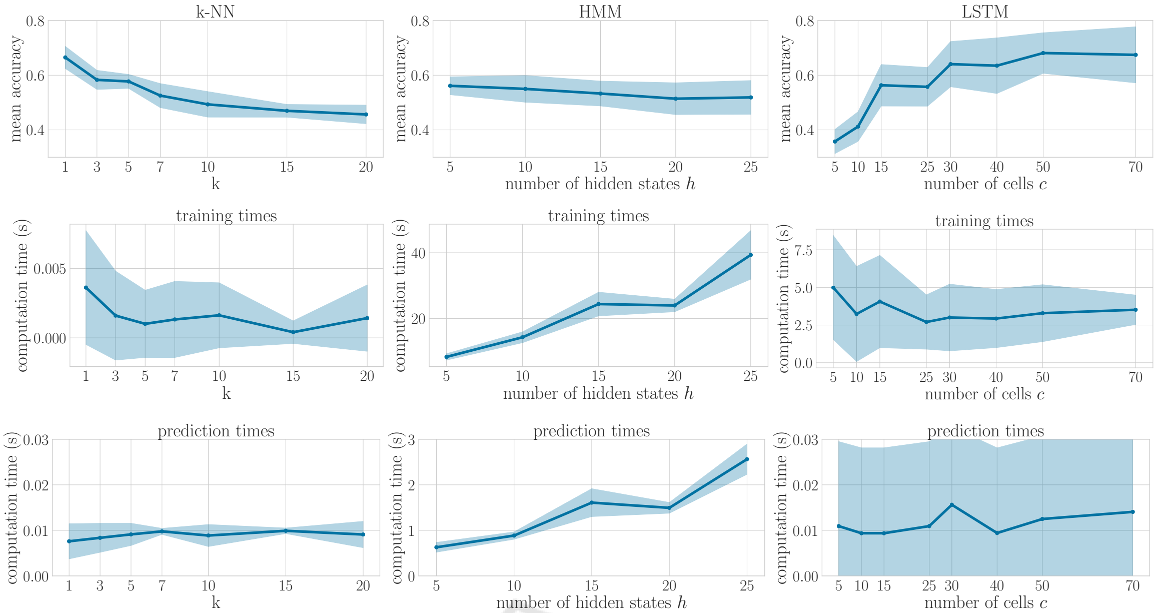

Figure 5: Classification results with limited training examples of the gesture data (experiment 1). Visualized is the classification with 10 examples per class. The top row shows the mean accuracy on the test data with different hyper-parameters of each classifier. Standard deviations are marked as colored areas. In the middle row, the training times are visualized, the bottom row shows the prediction times.

times in Fig. 5. All computations are run on a single core 3.7 GHz CPU without parallelization.

In Fig. 6a, the classification results with a number of examples per class in the training data between values from 1 to 20 is visualized. Hyper-parameters are set to $k = 1$ for k-NN, $c = 50$ for LSTM and $h = 5$ for HMM classification. With these configurations, the highest accuracies could be achieved. The 1-NN classifier slightly outperforms LSTM classification in this experiment. With HMM classification accuracies drop by $10-20\%$. Especially with very small training set sizes (≤ 10), HMM is clearly outperformed.

The evaluation results of the generalization to different subjects are shown in Fig. 6b. The LSTM network is the only approach that classifies the samples of subjects which are not part of the training data at a high mean accuracy around 90% if more than 4 examples of each class and each subject are used for training. The mean accuracy of 1-NN and HMM are below 50% in this experiment.

5.2 Classification of Stick-throwing Movements

Because of the lower complexity of the stick-throwing data, the validation on this data were performed with hyper-parameters set to $k \in [1,3,5,7,10]$ for k-NN, $h \in [2,5,10,15,20]$ for HMM, and $c \in [2,5,10,15,25]$ for LSTM.

With a maximum of 10 examples per class in the training data, best result is achieved with the LSTM classifier with $c = 25$ cells, leading to a mean accuracy of $94\%(\pm 0.03\%)$. k-NN with $k = 1$ reaches a mean accuracy of $88\%(\pm 0.03\%)$ and the HMM classifier has a mean accuracy of $70\%(\pm 0.04\%)$ with $h = 2$.

With these hyper-parameter settings, the classification accuracies with number of example per class in the training data between values from 1 to 20 is visualized in Fig. 7a. Like with the gesture data, LSTM and k-NN classification can deal well with very small training sets. With these two classifiers, an accuracy above 80% is reached with only 3-4 examples per class in the training data. With more examples per class, only small improvements can be observed. In comparison, the HMM classifier needs a mimimum of 15 examples per class to achieve the same result.

The results of the generalization capabilities of the classifiers is shown in the bottom graph of Fig. 7b. Again, LSTM generalizes best to new subjects with a mean accuracy above 80%, also with just 6 examples per class in the training data. In contrast to the gesture data, k-NN classifier also reaches good accuracies which are below the results of LSTM but still above 80%.

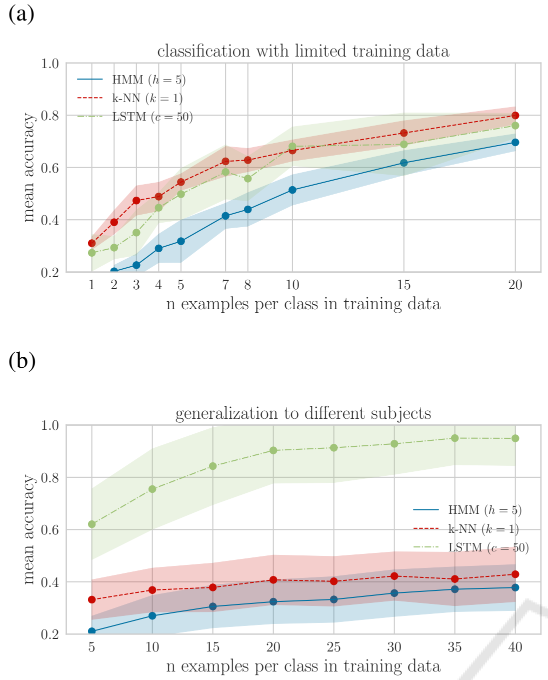

Figure 6: (a) Results of the classification of the gesture data with small training set sizes. (b) Results of the leave-one-subject-out cross-validation (experiment 2) on the gesture data.

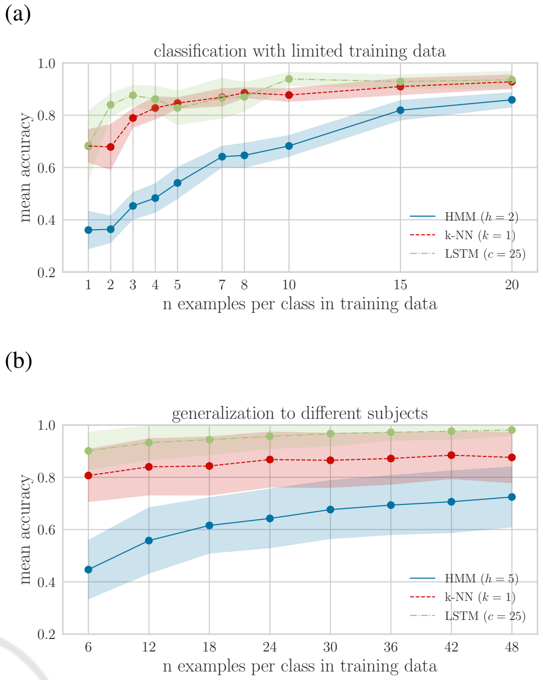

Figure 7: (a) Results of the classification of the stick-throwing data with small training set sizes. (b) Results of the leave-one-subject-out cross-validation (experiment 2) on the stick-throwing data.

6 DISCUSSION AND CONCLUSION

In the experiments in this paper, LSTM, HMM and k-NN were compared on movement data of different complexity with respect to classification with small training data sizes. Evaluations were performed on a gesture data set, which show large variations between subjects, as well as on a data set of stick-throwing movements. The throwing movements were simplified by segmenting the movement recordings into building blocks, which can be used, e.g. in robotics to equip a system with basic movements using LfD (Gutzeit et al., 2018).

The results show that with LSTM the best classification accuracies can be achieved. On the more heterogeneous gesture data set an accuracy of 80% is reached with 20 examples per class in the training data, on the more simple stick-throwing data 10 examples per class suffice for an accuracy above 90%. 1-NN also shows good classification results, but in contrast to LSTM it does not generalize well to new subjects on the gesture recordings. In this dataset, examples of the same gesture show a high variance between subjects and the clusters of the classes are more difficult to separate (see section 4.2). This makes generalization to new subjects difficult. On the much more simple stick-throwing data, which complexity is reduced by using automatic segmentation into building blocks, the examples of different subjects of the same movement class are more close and the movement classes are separated more clearly. However, 1-NN has fast calculation times, which makes 1-NN classification a clear alternative to the widely used neural network based approach, as it requires no hyper-parameter tuning and no architectures have to be defined. On both datasets, HMM requires more examples to model the demonstrations well enough for a good classification result and has higher computation times.

In conclusion, LSTMs give good results in the classification of different types of arm movements if the training is performed on very small training set sizes. It also generalizes to new subjects in the performed experiments. However, this has to be interpreted with caution, as this is highly dependent on the variations in the examples seen in the training data. If the data is simple, like the stick-throwing data analyzed in this paper, 1-NN is a clear alternative to LSTM. It requires no hyper-parameter tuning and has faster calculation times on small datasets. This strengthens our previous experiments on classification of manipulation building blocks using 1-NN (Gutzeit et al., 2019b). While the LSTM network performs better on data with higher inter-subject variations, this approach as well as HMM based classification cannot express their superior capabilities on sequenced data in the classification of building blocks of human arm movements.

For future work, a more detailed analysis of the influence of the segmentation into building blocks to reduce the complexity of the data, as well as the insights of human movement generation that can be inferred from this, would be of interest. These insights could help, e.g., to improve the generation of robotic behavior based on human examples to generate more flexible robotic systems.

ACKNOWLEDGEMENTS

This work was supported through two grants of the German Federal Ministry for Economic Affairs and Energy (BMWi, FKZ 50 RA 1703 and 50 RA 2023).

REFERENCES

Aarno, D. and Kragic, D. (2008). Motion intention recognition in robot assisted applications. *Robotics and Autonomous Systems*, 56:692–705.

Argall, B. D., Chernova, S., Veloso, M., and Browning, B. (2009). A survey of robot learning from demonstration. *Robotics and Autonomous Systems*, 57(5):469–483.

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer-Verlag New York, Inc.

Borghi, G., Vezzani, R., and Cucchiara, R. (2016). Fast gesture recognition with Multiple Stream Discrete HMMs on 3D skeletons. *Proceedings - International Conference on Pattern Recognition*, pages 997–1002.

Gutzeit, L., Fabisch, A., Otto, M., Metzen, J. H., Hansen, J., Kirchner, F., and Kirchner, E. A. (2018). The BesMan Learning Platform for Automated Robot Skill Learning. *Frontiers in Robotics and AI*, 5.

Gutzeit, L., Fabisch, A., Petzoldt, C., Wiese, H., and Kirchner, F. (2019a). Automated Robot Skill Learning from Demonstration for Various Robot Systems. In Benzmüller, C. and Stuckenschmidt, H., editors, *KI 2019: Advances in Artificial Intelligence, Conference Proc.*, volume LNAI 11793, pages 168–181. Springer.

Gutzeit, L., Otto, M., and Kirchner, E. A. (2019b). Simple and robust automatic detection and recognition of human movement patterns in tasks of different complexity. In *Physiological Computing Systems*. Springer.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Liu, J., Shahroudy, A., Xu, D., Kot Chichung, A., and Wang, G. (2017). Skeleton-Based Action Recognition Using Spatio-Temporal LSTM Network with Trust Gates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(12):3007–3021.

Patsadu, O., Nukoolkit, C., and Watanapa, B. (2012). Human gesture recognition using Kinect camera. *Computer Science and Software Engineering (JCSSE), 2012 International Joint Conference on*, pages 28–32.

Poppe, R. (2010). A survey on vision-based human action recognition. *Image and Vision Computing*, 28(6):976–990.

Senger, L., Schröer, M., Metzen, J. H., and Kirchner, E. A. (2014). Velocity-based Multiple Change-point Inference for Unsupervised Segmentation of Human Movement Behavior. In *Proccedings of the 22th International Conference on Pattern Recognition (ICPR2014)*, pages 4564–4569.

Shi, Y., Tian, Y., Wang, Y., and Huang, T. (2017). Sequential Deep Trajectory Descriptor for Action Recognition with Three-Stream CNN. *IEEE Transactions on Multimedia*, 19(7):1510–1520.

Stefanov, N., Peer, A., and Buss, M. (2010). Online intention recognition for computer-assisted teleoperation. In *Proceedings - IEEE International Conference on Robotics and Automation*, pages 5334–5339.

van der Maaten, L. and Hinton, G. (2008). Visualizing Data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605.

Wang, Y., Yao, Q., Kwok, J. T., and Ni, L. M. (2020). Generalizing from a Few Examples: A Survey on Few-shot Learning. *ACM Computing Surveys*, 53(3).