LSTM Self-Supervision for Detailed Behavior Analysis

Biagio Brattoli[1]*, Uta Büchler[1]*, Anna-Sophia Wahl[2], Martin E. Schwab[2], Björn Ommer[1]
[1] HCI / IWR, Heidelberg University, Germany
[2] Department of HST, ETH Zurich, Switzerland
{biagio.brattoli,uta.buechler,bjoern.ommer}@iwr.uni-heidelberg.de,
{wahl,schwab}@hifo.uzh.ch

Abstract

Behavior analysis provides a crucial non-invasive and easily accessible diagnostic tool for biomedical research. A detailed analysis of posture changes during skilled motor tasks can reveal distinct functional deficits and their restoration during recovery. Our specific scenario is based on a neuroscientific study of rodents recovering from a large sensorimotor cortex stroke and skilled forelimb grasping is being recorded. Given large amounts of unlabeled videos that are recorded during such long-term studies, we seek an approach that captures fine-grained details of posture and its change during rehabilitation without costly manual supervision. Therefore, we utilize self-supervision to automatically learn accurate posture and behavior representations for analyzing motor function. Learning our model depends on the following fundamental elements: (i) *limb detection based on a fully convolutional network is initialized solely using motion information,* (ii) *a novel self-supervised training of LSTMs using only temporal permutation yields a detailed representation of behavior, and* (iii) *back-propagation of this sequence representation also improves the description of individual postures. We establish a novel test dataset with expert annotations for evaluation of fine-grained behavior analysis. Moreover, we demonstrate the generality of our approach by successfully applying it to self-supervised learning of human posture on two standard benchmark datasets.*

1. Introduction

Motor behavior is the main form of an individual to express and interact with the environment. Analysis and precise quantification of motor kinematics, therefore, provide a detailed, non-invasive understanding of functional deficits of the sensorimotor cortex. Furthermore, videos of behavior recorded during long-term studies on the recovery after neurological diseases provide an easily available, rich source of information to evaluate and adjust drug application and rehabilitative paradigms. The main bottleneck is presently that all analysis of skilled motor function depends on a time-intensive, error-prone, and costly manual evaluation of behavior, e.g., by aggregating a large set of subtle characteristics of limb posture and its deformation over time [1]. Consequently, this detailed behavior representation required for studying skilled motor functions goes far beyond a trajectory analysis [23] as can also be seen from Fig. 6. Thus, there is a dire need for an automatic evaluation of subtle differences in behavior and the underlying postures [4]. Given the large amounts of available unlabeled video data, we seek a self-supervised method that can learn *(i)* to detect the limb, which is performing skilled motor function, *(ii)* to represent its posture so that it captures subtle differences due to impairment while being invariant to variations between animals, *(iii)* and to compare behavior sequences with another.

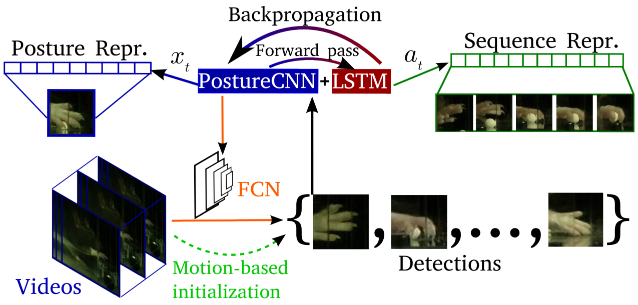


Figure 1: Overview of our self-supervised approach for posture and sequence representation learning using CNN-LSTM. After the initial training with motion-based detections we retrain our model for enhancing the learning of the representations.

Our specific setting involves long-term recordings of rats recovering from a large stroke in the sensorimotor cortex

*Indicates equal contribution

[27] (the second leading source of disability worldwide) and performing skilled forelimb grasping. The only available information for training are videos recorded before and after stroke, where even the healthy animals show a substantial number of failed grasps due to the complexity of the task. We address challenges *(i)-(iii)* jointly by combining a CNN for individual postures with an LSTM for behavior sequences which train themselves without the need of annotated grasping sequences. Given weak initial candidate detections of grasping paws obtained using motion information, a CNN is trained to separate paws from clutter. Unrolling the fully convolutional layers of this model, we obtain a fully convolutional network (FCN [37]) for detecting paws. Moreover, due to the absence of posture annotations we will also utilize this CNN model as an implicitly learned, initial representation of posture. To further improve this representation we move from posture to behavior sequences. Therefore, the CNN for individual postures is directly linked to a recurrent network (LSTM) for behavior, indirectly optimizing the posture representation using the surrogate task of behavior learning through sequence ordering. Although this task of training an LSTM on original sequences against permuted ones sounds more difficult, we can now tap the large amounts of unlabeled videos by self-supervision. Bootstrap retraining then improves detections which in turn enhance the learning of behavior and as a result the individual posture representation, cf. Fig 1. Finally, we use multiple instance learning (MIL)[3, 5] to train a classifier to discover the subtle differences between healthy and impaired grasping behavior.
In the experimental evaluation on a novel test dataset for detailed behavior analysis our approach compares favorably against expert manual evaluation of behavior that has been established in neuroscience. Moreover, we show the generality of our method by successfully applying it to two standard benchmark datasets for human pose estimation (Olympic Sports and Leeds dataset [32, 22]), where it improves upon the state-of-the-art for unsupervised pose analysis.

2. Related work

In the age of big data, problems have shifted from lacking training data to now having lots of it but lacking tedious manual annotations. Deep learning, which benefits from large volumes of training data, has therefore spurred new interest in unsupervised techniques and especially self-supervision has recently shown great potential [11, 30, 33, 36, 40]. Therefore, surrogate tasks are created to indirectly learn a representation that can be transferred to the original problem. [11], for instance, pre-train a network using the spatial context in images without requiring labels. Misra et al. [30], on the other hand, utilize spatiotemporal information by verifying the temporal order of three frames using a triplet siamese network. These approaches learn a single frame representation, which is not sufficient for our challenge of learning similarities between behavior sequences. Thus, we combine learning of single frame posture and kinematics for their mutual benefit in an LSTM, which we train on permutations of entire sequences.

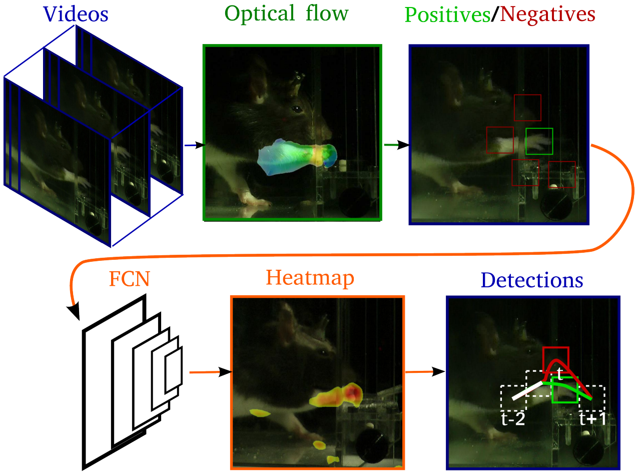


Figure 2: Visualization of our detection system, which uses optical flow for initial positive samples and random negatives to train an FCN for extracting candidate regions before applying temporal smoothing (bottom right).

An initial problem of kinematic analysis concerns tracking. Tracking approaches can be grouped into methods based on visual object [41, 25] or model transfer [20, 26]. The absence of initialization, frequent occlusion, high variance in appearance, and motion blur makes limb tracking a very challenging task. Also tracking-by-detection [17] is not applicable as we lack an initialization of the object we should track. Therefore, detection of limbs and the overall problem of learning behavior need to be addressed, jointly.

Pose estimation is traditionally tackled by detecting body parts [35, 39]. Lately, extra information is used to improve the results, like the relative position between parts[34] or the appearance of the subject in a video [9]. Our approach does not require any model of the body or its parts. Rather, characteristic pose information is extracted directly from the image using a similarity between poses that has been learned by LSTM self-supervision.

Although behavior analysis is of crucial importance for numerous vision tasks and there has been significant progress on pose [8, 39, 35, 6] and action classification [38, 30, 36, 7], detailed kinematic studies, which are required in many neuroscientific studies, have still not been tackled sufficiently. Therefore, manual analysis of large volumes of video recordings is still the usual practice when investigating subtle changes of behavior [2, 31, 16]. Even

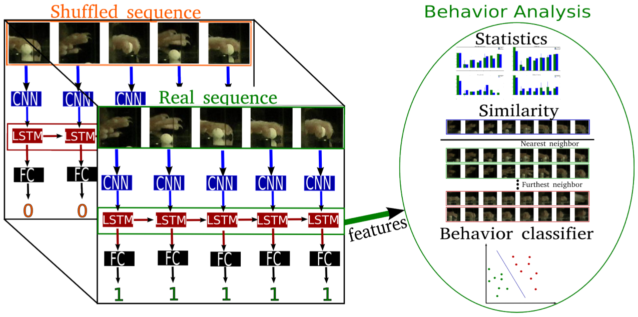


Figure 3: Visualization of the LSTM self-supervised training. As input we use real and permuted sequences to learn a representation of posture and behavior sequences. These are utilized for further analysis of behavior.

for the simpler setting of trajectory analysis, heavily supervised methods are being used [23]. Another approach has been to use specialized equipment [14, 18] and avoid the complexities of behavior analysis altogether by merely counting if a limb touches or fails particular tactile sensors [15]. However, this cannot reveal in what way skilled motor function is impaired.

3. A Self-Supervised Approach to Behavior Analysis

Now we present an approach for performing long-term behavior studies without requiring tedious manual supervision. In particular, this method uncovers the details of restoration of skilled motor function during recovery after brain lesions. In our concrete setting of grasping rats we jointly learn to detect paws, to represent their postures, and to compare complete grasping sequences from unlabeled video sequences to analyze them.

3.1. Detection

The initial challenge of any behavior analysis is detecting the limbs—in case of skilled forelimb function the interest is on the acting hand. Finding and tracking a rat paw during grasping is challenging for a number of reasons. There is no initialization for tracking provided and due to frequent occlusions by other body parts (other paw, arm, nose, etc.) detections are frequently lost. Moreover, paws are small, furry, fast-moving (implying large displacement between successive frames and motion blur due to limited illumination of nocturnal rodents that are distracted by intense light), and appearance varies significantly between subjects as does shape between different hand postures. Learning a representation and detector for paws with these large variations in shape and appearance is therefore demanding—especially since we do not require laborious manual annotations of paws. Therefore, we follow a sequential bootstrapping procedure to train a CNN-based hand detector in an iterative manner, initializing it with motion information to start with the easy to extract paws first and then consecutively learning more complex ones. We initially extract a set of candidate paw regions by computing optical flow [28] and decomposing frames with [42] into a low-rank background model and a sparse set of connected foreground pixels, thus finding strongly moving paws. In addition to these positive samples we add hard negatives randomly sampled from locations around the positives to then train a CNN (AlexNet [24], trained with stochastic gradient descend with cross-entropy loss) to separate both classes. This *PostureCNN*$_0$ is then turned into a fully convolutional network (*FCN*$_0$) by reshaping the last 3 fc-layers. A deconvolutional layer is not necessary since we do not need pixel-accurate segmentation. Paws are then extracted by taking the 5 strongest candidate detections from the *FCN*$_0$ scoremap and then selecting the best one by temporal smoothing, i.e., fitting a polynomial to ten consecutive frames and choosing the smoothest trajectory. Fig. 2 summarizes this procedure. This new detector has better performance than the initial motion detections as shown in the experimental section 4. From now one we refer to a detection as d_t where t is the index of a frame.

3.2. Self-supervised Learning of Postures and Behavior

To facilitate paw detection, the *PostureCNN*$_0$ has implicitly learned a representation of paws that we now use as initialization for training a more accurate model of posture. After all a model that separates paws from clutter will not suffice to capture the fine-grained posture similarities needed for analyzing degradation of skilled motor function. Let $\phi(\cdot)$ denote the fc6 output of *PostureCNN*$_0$, then $x_t = \phi(d_t)$ is the resulting initial posture representation for a paw detection d_t. Since tedious annotations of posture are not available, we cannot simply train a multi-class classifier to separate them, but instead employ self-supervised training. Therefore, we move from individual postures to behavior sequences to utilize temporal context and learn the detailed changes in posture during grasps. A recurrent neural network links the posture x_t in consecutive frames by means of hidden states h_t and a non-linear activation function σ,

$$\begin{aligned} h_t &= \sigma(W_h x_t + U_h h_{t-1} + b_h) \\ a_t &= \sigma(W_a h_t + b_a)\ . \end{aligned} \tag{1}$$

By learning the parameters W, U, b we obtain an output representation a_t. The model is implemented as a long short-term memory (LSTM) [12]. To represent behavior we stack the conv1 to fc6 layers of the *PostureCNN*$_0$ with the

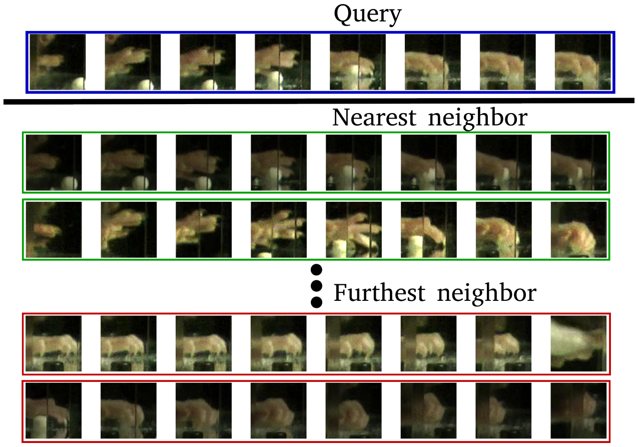


Figure 4: 2 nearest and 2 furthest neighbors of a query behavior sequence given the learned behavior representation.

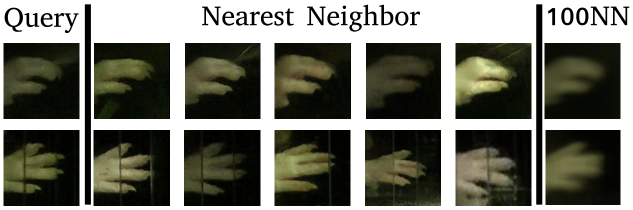


Figure 5: Applying the posture representation to measure similarity: 5 nearest neighbors and average of the 100 nearest neighbors for two query frames.

LSTM and a final fully-connected layer (FC) on top (*CNN-LSTM*$_1$). During training the gradient is back-propagated from the top fc classifier down to conv1, jointly updating the behavior in the LSTM layer and all intermediate layers of the posture representation. A visual summary of this joint training of posture and behavior is shown in Fig. 1.

For training the presented model without any provided annotations, we implement a novel self-supervised learning method based on a surrogate task. We teach the network to distinguish between real and randomly permuted sequences $s_t = [d_t, d_{t+1}, \cdots, d_{t+l-1}]$, where l is the length of the sequence, in a binary classification scenario. For training, mini-batches D_i are composed as

$$\begin{aligned} D_i &= [s_t^{i_1}, \pi(s_t^{i_1}), s_t^{i_2}, \pi(s_t^{i_2}), \cdots, s_t^{i_n}, \pi(s_t^{i_n})] \\ L_i &= [y_1, y_2, \cdots, y_{2n}], \text{ with } y_k = k \bmod 2 \end{aligned} \tag{2}$$

where y_k is the label of sequence k in D_i, $\pi(\cdot)$ is a random permutation of the frames in the sequence, and $2n$ is the number of sequences in batch D_i. Distinguishing between a real sequence and its permutation is challenging, since the model needs to learn subtle details of posture together with their change over time. Moreover, a positive and the respective negative sequence are identical except for the order in which postures appear, thus forcing the network to disregard background, differences between animals, or lighting changes and rather focus on the change of postures during grasping. Our surrogate task is perfectly suited for learning behavior and postures, while establishing crucial independence properties, as well as the only feasible solution for our problem due to the missing labels. Given a sequence s_t we now define a sequence representation a_t (from Eq. 1) as the output of the LSTM layer from the learned *CNN-LSTM*$_1$. Moreover, we obtain an updated posture representation $x'_t = \phi'(d_t)$, since the fc6 layer has been retrained compared to the original *PostureCNN*$_0$. The learned representation a_t is used for behavior analysis, predominantly for computing the distance measure between sequences by calculating the cosine similarity. Furthermore we can solve classification tasks on sequences by training a linear classifier on a_t. Fig. 3 describes our self-supervised approach for learning sequence (and posture) representations and its application. The building blocks for training are the CNN (*PostureCNN*), the LSTM layer and the FC layer. Note that the FC layer is only necessary for training.

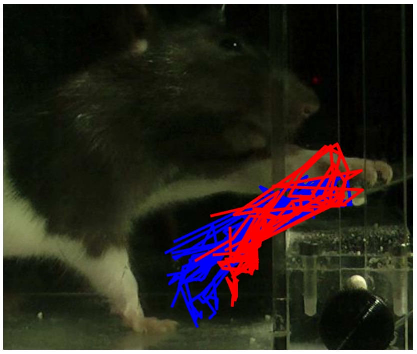


Figure 6: Side view of a grasping rat. Trajectories of 50 good (blue) and 50 impaired (red) grasps are superimposed. Evidently, trajectories do not suffice to capture impairment of behavior.

This paw representation can in turn be used for replacing the initial motion-based paw detections in Sect. 3.1. Thus we can bootstrap our approach and retrain the *PostureCNN*$_0$ (denoted *PostureCNN*$_1$) to obtain the improved detector (*FCN*$_1$). The sequence model *CNN-LSTM*$_2$ is then obtained by fine-tuning *CNN-LSTM*$_1$ on the new detections gained from *FCN*$_1$. In our experiments we have observed rapid convergence after two rounds and we analyze the gain of this bootstrapping in Sect. 4.

3.3. Grasp analysis using MIL approach

Let us now use the learned behavior representation a_t (from Eq. 1) to discover what subtle differences make a

Models	Accuracy(%)
OpticalFlow [28]	40.2
FCN$_0$	58.0
FCN$_1$	81.4
FCN$_2$	**82.1**

Table 1: Accuracy of the detection obtained by optical flow based initialization, after one round of training (*FCN*$_0$), and after two (*FCN*$_1$) and three rounds of re-training (*FCN*$_2$).

grasp fail. Besides discriminating between successful and unsuccessful grasps this analysis will highlight how motor function improves over the course of rehabilitation. For training we will only use grasping videos v recorded before and right after the animal incurred a photothrombotic stroke [27]. In the latter case all grasps are impaired (negatives), whereas beforehand, due to the difficulty of the task, there are good (positives) but also around 20% corrupted grasps. Grasps from later stages of rehabilitation are not considered for training since their outcome is unknown and needs to be inferred. We address the learning problem using a Multiple-Instance Learning (MIL) approach [3]. We assemble bags B_v of a_t^v from videos v, $B_v = \{a_t^{v_1}, a_t^{v_2}, \dots\}$, with label Y_v. A bag B_v has a negative label if all samples of the bag are negatives, $Y_v = -1 \Leftrightarrow \forall i : y_{v_i} = -1$. It is positive if at least one of the samples is positive $Y_v = 1 \Leftrightarrow \exists i : y_{v_i} = 1$. MIL tries to find the best discriminative classifier and, at the same time, infers the right label for each sample. The MIL approach then iteratively *(i)* trains a classifier (we use linear soft-margin SVM) and *(ii)* again imputes labels for each sequence in a bag using the classifier. Without labels of individual sequences the model then learns the characteristics of a failed grasp as discussed in Sect. 4.1.3.

4. Experiments

In this chapter we are going to describe our experiments in detail and evaluate our approach quantitatively and qualitatively, first on a biomedical dataset and afterwards on two standard benchmark datasets for human pose estimation. All deep networks in our experiments are implemented using the CAFFE framework [21].

4.1. Analyzing Skilled Motor Function

We have established a new dataset of rats performing skilled forelimb action (single pellet grasping) before and during the recovery from a photothrombotic stroke. The dataset consists of 242 videos of 26 rats with an average length of 5 minutes per video, which is in total $\sim$ 20 hours recorded at 50 fps or 3.7M frames. Recording sessions were 0 (Baseline), 2, 7, 14, 21, 28, and 35 days after the stroke. Animals come from four different cohorts with different treatment paradigms: neuronal growth promoting antibody drug [27] (1) without and (2) with physical training, (3) physical training without neuronal stimulation, and (4) a cohort with no treatment at all.

Models	Accuracy(%)
AlexNet [24]	65.3
PostureCNN$_0$	72
PostureCNN$_2$	**85.6**

Table 2: Evaluation of posture representation using the benchmark test set for posture similarity.

4.1.1 Detection

We obtain $\sim$$15,000$ initial detections using the optical flow based initialization. These detections serve as positive samples for training the *PostureCNN*$_0$ (training mini-batch size: 256 samples). Every detection is augmented 10 times with random scaling and translation. This yields a total training set size of $300,000$ samples (including negatives). Bootstrap retraining of the CNN-LSTM sequence representation and the FCN paw detections finally converges after two iterations. To evaluate performance, we established a small test set of manually labeled paw locations from different videos. Table 1 shows the detection accuracy of successive rounds of the bootstrap retraining (detections are counted as correct if their intersection over union with the groundtruth is $\geq 50\%$).

4.1.2 Evaluating Behavior and Posture Representation

The LSTM layer is initialized with random weights and 512 hidden nodes. We tested our model with different numbers of hidden nodes, but more than 512 nodes did not enhance the final accuracy. We use $n = 12$ sequences per batch and add a randomly re-ordered version for each. Our training set is composed of around $40,000$ densely extracted sequences. The network is trained for classification using stochastic gradient descend with cross-entropy loss.

We evaluate the learned representations using a posture (individual frames) and a behavior (sequences) benchmark,

Models	Accuracy(%)
Max frame similarity	74.1
Avg frame similarity	75.9
DTW[10]	76.8
ClusterLSTM	64.0
CNN-LSTM$_2$	**80.5**

Table 3: Evaluation of behavior representation using the benchmark test set for sequence similarity.

which were manually labeled by neuroscientists. Both benchmarks are composed of reference frames/sequences and 10 similar and 10 dissimilar manually selected samples have been added for each reference. We use $I_f = 30$ reference elements for evaluating the posture representation and $I_s = 22$ for the sequence evaluation. In total this test set consists of 4326 frames. Given the learned representation, we measure the similarity of the reference to the 10 related and 10 unrelated samples to order them.

Tab. 2 shows the resulting accuracies for single frame posture similarity obtained by standard *AlexNet*, our $\mathit{PostureCNN}_0$, and the final representation obtained by two iterations, $\mathit{PostureCNN}_2$. Note that the joint training of behavior and postures substantially improves the representation of individual postures. In Tab. 2 and 3 we omit the result of the intermediate iteration ($\mathit{PostureCNN}_1$,$\mathit{CNN\text{-}LSTM}_1$), since performance differed only marginally due to convergence of learning (as indicated in Table 1).

We compare the accuracy of behavior representation achieved by our LSTM ordering task with *Dynamic Time Warping*[10] (DTW), a direct stacking of single frame posture representations and an LSTM baseline model (*clusterLSTM*) based on sequence clustering. In case of stacking, similarities between all frames of both sequences are computed to then use either the maximum or the average for comparison. *ClusterLSTM* is the LSTM network trained on a multi-class classification task rather than our proposed self-supervised ordering task of Eq. 2. Therefore, we create clusters of sequences using Dynamic Time Warping as distance measure between the sequences and train the network to separate different clusters from another. Tab. 3 shows that our LSTM ordering task improves upon all the other approaches. The weak performance of *clusterLSTM* underlines that training a behavior representation on discrete groups of sequences is not suited to learn fine-grained behavior similarities.

For a qualitative evaluation we show in Fig. 5 and 4 nearest neighbors of a query frame and a sequence, respectively. The frame/sequence similarity is based on cosine distances of the fc6 representation of our joint approach for the frame representation and the LSTM-features for the sequence representation. Note that the nearest neighbors in Fig.5 span different animals thus portraying invariance to differences between videos.

4.1.3 Behavior analysis

After previously evaluating our posture and behavior representation, we now compare the approach against the manual evaluation setup used by neuroscientists and we show its potential for further applications.

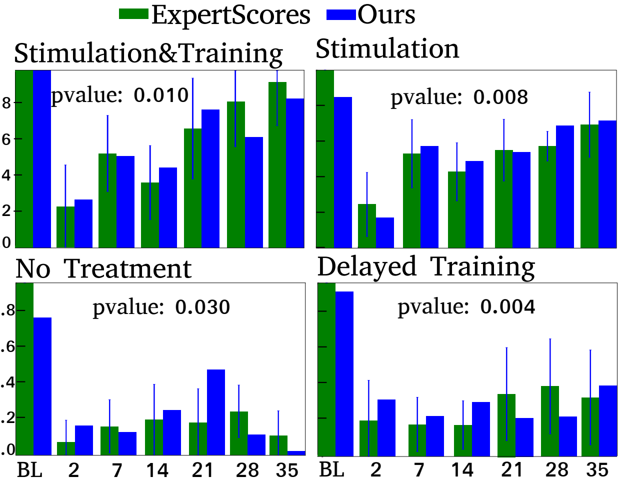


Figure 7: Green: Expert scores [1] of grasping fitness for all animals of a specific cohort during an experimental session, which is designated by the time in days since the stroke. BL are the baseline recordings of healthy animals before stroke. Blue: Grasping fitness as predicted by our self-supervised approach. Agreement between both scores is measured by the p-value. See text for details.

Predicting the Fitness of Skilled Motor Function: For neuroscientists the primary goal of behavior analysis is to discover the degree of impairment of subjects that are performing skilled motor function. For the task of single pellet grasping, a standard protocol has been proposed [1] to judge grasping fitness. Experts assess grasping by scoring ten criteria including the pronation and supination of the paw (its turning) and the opening and closing of the digits. Averaging these scores then yields an indicator for the fitness of the motor function. Rather than trying to replicate the individual decisions that experts make, we propose to circumvent this tedious manual analysis by directly mapping sequences to a final fitness score. However, since there are no labeled training sequences, we utilize the MIL approach from Sect. 3.3 to learn how healthy the behavior appears. Fig. 7 shows predicted and expert scores for all animals of a cohort based on the behavior in each experimental session. Our predicted scores are calibrated (scaling, additive shift) to have the same range of values as the expert scores. We measure the agreement between both scores using the p-value of the two-tailed t-statistic of a linear regression between the scores (null hypothesis is that our score does *not* predict the expert scores). For the four cohorts we obtain p-values in the range of $p = 0.004$ to $p = 0.03$ indicating that the null hypothesis can be savely rejected. Over the coarse of rehabilitation our approach nicely discovers the improvement in the two treatment cohorts at the top of Fig. 7 compared to no/delayed treatment at the bottom.

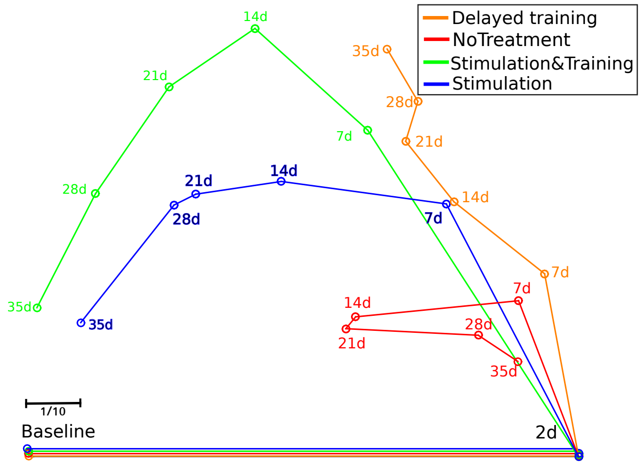


Figure 8: Visualizing the recovery process. All kinematics of an experimental session are compared against impaired behavior 2 days post stroke and against good behavior at baseline (after removing bad ones using the MIL approach). The scalebar shows a tenth of the distance between the two references, baseline and 2d. The two treatment cohorts (blue and green) show good recovery in contrast to the other two that also change in kinematics, but not for good.

Behavior Kinematics vs. Trajectory Analysis: Since trajectories are easier to obtain, much of previous work in biomedical imaging (e.g. [23]) has settled for a mere trajectory analysis instead of the detailed analysis of behavior kinematics utilized by [1, 4] and in our approach. In the following we are going to evaluate the aptitude of trajectories. Therefore, we stack the paw locations for each grasping sequence instead of their appearance and again train a MIL SVM classifier to separate baseline recordings from 2 days post stroke. The resulting classifier only achieves chance-level performance, indicating that trajectories are not sufficient to identify subtle differences in behavior. And indeed, plotting 50 good (blue) and 50 impaired grasps (red), Fig.6, reveals that both sets are heavily overlapping.

Rehabilitation Analysis: From long-term recordings over rehabilitation, behavior analysis can also reveal the subtle changes in motor function during recovery. Obviously there are thousands of trials between an initially impaired and the finally good sequence. We relate the behavior to a large set of healthy baseline (BL) kinematics (after removing bad ones using the MIL procedure from Sect. 3.3) as well as impaired samples from 2 days post stroke (2d). Our LSTM-based behavior representation thus provides distances of motor function at any date to BL and 2d, which then define a two dimensional summary of the course of rehabilitation, displayed in Fig. 8 using triangulation for the 2D plot. The figure not only shows, if the animal behavior gets closer again to their original state at baseline, it also reveals cases of unsuccessful recovery where behavior digresses from 2d, however without becoming more similar to baseline, e.g. delayed training.

4.2. Self-supervised Human Pose Estimation

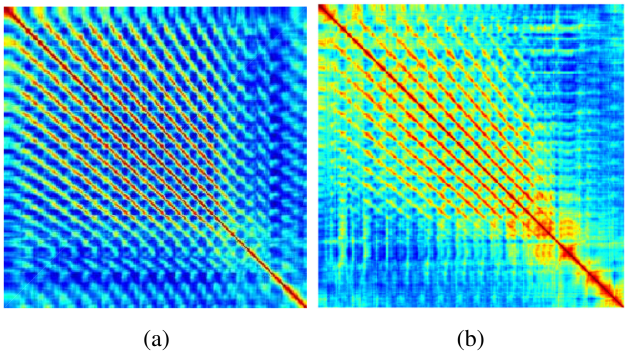
We now evaluate the generality of our approach by investigating its applicability to human pose analysis without requiring annotations. We perform experiments on two standard benchmark datasets. We first train and evaluate on the standard Olympic Sports dataset [32]. We then show that our approach can effectively transfer this learned representation to another standard benchmark, the Leeds Sports dataset [22].

Olympic Sports Dataset: The Olympic Sports dataset [32] comprises 16 different sport activities with a total of 525 clips and 113, 516 frames. To initialize our *postureCNN* we utilize the filters of the powerful CliqueCNN model [8], which has shown competitive performance on pose analysis with unlabeled data. Then we run our self-supervised training procedure as explained in Sect. 3.2 to learn behavior and improve the posture representation. To be comparable we use the same experimental setup [1] as in [8] with around 1033 test examples and having to order 20 similar and dissimilar samples to each of the examples.

Category	HOG-LDA [19]	Ex. SVM [29]	Ex. CNN [13]	Alex net [24]	Clique CNN [8]	Ours
Basketball	0.51	0.63	0.58	0.55	0.70	**0.75**
Bowling	0.57	0.63	0.58	0.55	0.85	**0.87**
Clean&Jerk	0.61	0.71	0.58	0.62	0.81	**0.85**
Discus Thr.	0.42	0.76	0.56	0.59	0.65	**0.68**
Diving 10m	0.42	0.54	0.51	0.57	0.70	**0.76**
Diving 3m	0.50	0.57	0.52	0.66	0.76	**0.84**
HammerThr.	0.62	0.64	0.51	0.66	0.82	**0.88**
High Jump	0.64	0.76	0.59	0.62	0.82	**0.87**
Javelin Thr.	0.71	0.72	0.57	0.74	**0.85**	**0.85**
Long Jump	0.60	0.69	0.57	0.71	0.78	**0.85**
Pole Vault	0.59	0.64	0.60	0.64	0.81	**0.83**
Shot Put	0.51	0.67	0.52	0.70	0.75	**0.76**
Snatch	0.64	0.76	0.59	0.67	0.84	**0.89**
TennisServe	0.70	0.75	0.64	0.71	0.84	**0.87**
Triple Jump	0.63	0.65	0.58	0.65	0.80	**0.83**
Vault	0.59	0.71	0.63	0.68	0.81	**0.86**
Mean	0.58	0.67	0.56	0.65	0.79	**0.83**

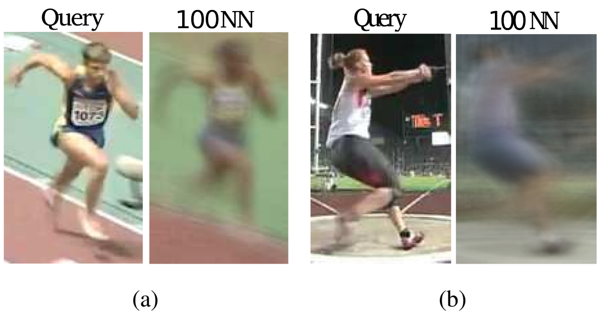
Table 4: Average AUC of all categories of the Olympic Sports dataset using the state-of-the-art and our approach.

[1] https://asanakoy.github.io/cliquecnn/


(a)

(b)

Figure 9: Similarity Matrices of the category *Long Jump* using (a) our approach and (b) CliqueCNN [8]. Rows/columns are individual frames.



(a)

(b)

Figure 10: Applying the posture representation obtained by our self-supervised LSTM training to find similar samples: Averaging the 100 nearest neighbors of a query frame from category (a) *Long Jump* and (b) *Hammer Throw*.

Tab. 4 shows the average AUC using HOG-LDA[19], Exemplar-SVMs[29], Exemplar-CNN[13], CliqueCNN[8], and our approach. We consistently outperform all previous methods and achieve an average gain of 4% over the best performing previous method, CliqueCNN. Evidently, our self-supervised LSTM-based sequence ordering task captures more subtle structures, which becomes apparent when comparing the learned pose similarities of our approach Fig. 9a with that of the state-of-the-art Fig. 9b. The stripe pattern highlights the reoccurring postures of gait cycles in *Long Jump* and it is much more clearly pronounced than in the previous approach. Since our model has learned detailed similarities, it can find a large number of consistent nearest neighbors to a query frame and then produce averages that still capture the essence of the pose. Fig. 10a and 10b average over 100 nearest samples.

Leeds Sports Dataset: The Leeds Sports dataset [22] contains 2000 pose annotated images. Since there are only static images but no sequences as would be required for Sect. 3.2, we transfer the previously trained model from Olympic Sports and directly test on the Leeds Sports benchmark. For evaluation we follow the standard protocol and measure the Percentage of Correct Parts (PCP). In Tab. 5 we show the PCP acquired by HOG-LDA, Alexnet, CliqueCNN, our approach, two fully supervised method (Pose Machines [35] and DeepCut [34]), and using the ground-truth (GT) similarities. The GT indicates an upper bound on the performance we can achieve by an unsupervised approach that finds nearest training samples to query frames, but that is not trained on keypoints. Therefore, for each query we here select the nearest neighbor using its keypoints and measure the PCP between the groundtruth keypoint annolation of the nearest neighbor and the query. Compared to the currently best unsupervised method (CliqueCNN) we improve by 3.1%. Achieving this gain without fine-tuning on the target dataset shows that our approach can nicely generalize.

Parts	HOG LDA [19]	Alex net [24]	Clique CNN [8]	Ours	Pose Mach. [35]	Deep Cut [34]	GT
Torso	73.7	76.9	80.1	**82.4**	88.1	96.0	93.7
U.legs	41.8	47.8	50.1	**53.3**	79.0	91.0	78.8
L.legs	39.2	41.8	45.7	**48.0**	73.6	83.5	74.9
U.arms	23.2	26.7	27.2	**30.9**	62.8	82.8	58.7
L.arms	10.3	11.2	12.6	**16.0**	39.5	71.8	36.4
Head	42.2	42.4	45.5	**48.9**	80.4	96.2	72.4
Mean	38.4	41.1	43.5	**46.6**	67.8	85.0	69.2

Table 5: PCP measure (observer-centric) of the Leeds Sport dataset using all mentioned approaches.

5. Conclusion

We have presented a self-supervised procedure for learning detailed posture and behavior from large amounts of unlabeled video sequences. A CNN for pose representation is interlinked with an LSTM for behavior and an FCN hand detector. These components mutually benefit from another, due to a joint training procedure. We have proposed a temporal ordering task for learning the LSTM and improving single frame posture that does not require any fine-tuning. Multiple instance learning then provides a non-parametric classifier, which can predict the recovery status and it compares favorably against a tedious manual approach followed by neuroscientists so far. Moreover, the approach has proven its wide applicability and shown competitive performance on standard benchmark datasets for human posture estimation.

Acknowledgement: We are grateful to the NVIDIA corporation for supporting our research, the experiments in this paper were performed on a donated Titan X GPU.

References

[1] M. Alaverdashvili and I. Q. Whishaw. A behavioral method for identifying recovery and compensation: hand use in a preclinical stroke model using the single pellet reaching task. *Neuroscience & Biobehavioral Reviews*, 37(5):950–967, 2013. 1, 6, 7

[2] B. Alstermark and L.-G. Pettersson. Skilled reaching and grasping in the rat: lacking effect of corticospinal lesion. *Arm and Hand Movement: Current Knowledge and Future Perspective*, 5(103):118, 2015. 2

[3] S. Andrews, I. Tsochantaridis, and T. Hofmann. Support vector machines for multiple-instance learning. In S. Becker, S. Thrun, and K. Obermayer, editors, *Advances in Neural Information Processing Systems 15*, pages 577–584. MIT Press, 2003. 2, 5

[4] B. Antic, U. Büchler, A. S. Wahl, M. E. Schwab, and B. Ommer. Spatiotemporal parsing of motor kinematics for assessing stroke recovery. In *Medical Image Computing and Computer-Assisted Intervention*. Springer, Springer, 2015. 1, 7

[5] B. Antic and B. Ommer. Robust multiple-instance learning with superbags. In *Asian Conference on Computer Vision*, pages 242–255. Springer, Springer, 2012. 2

[6] B. Antic and B. Ommer. Learning latent constituents for recognition of group activities in video. In *European Conference on Computer Vision*. IEEE, 2014. 2

[7] B. Antic and B. Ommer. Per-sample kernel adaptation for visual recognition and grouping. In *IEEE International Conference on Computer Vision*. IEEE, 2015. 2

[8] M. A. Bautista, A. Sanakoyeu, E. Sutter, and B. Ommer. Cliquecnn: Deep unsupervised exemplar learning. *NIPS*, 2016. 2, 7, 8

[9] J. Charles, T. Pfister, D. Magee, D. Hogg, and A. Zisserman. Personalizing human video pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3063–3072, 2016. 2

[10] T. Darrell and A. Pentland. Space-time gestures. In *Computer Vision and Pattern Recognition, 1993. Proceedings CVPR'93., 1993 IEEE Computer Society Conference on*, pages 335–340. IEEE, 1993. 5, 6

[11] C. Doersch, A. Gupta, and A. A. Efros. Unsupervised visual representation learning by context prediction. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1422–1430, 2015. 2

[12] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2625–2634, 2015. 3

[13] A. Dosovitskiy, J. T. Springenberg, M. Riedmiller, and T. Brox. Discriminative unsupervised feature learning with convolutional neural networks. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 766–774. Curran Associates, Inc., 2014. 7, 8

[14] D. J. Ellens, M. Gaidica, A. Toader, S. Peng, S. Shue, T. John, A. Bova, and D. K. Leventhal. An automated rat single pellet reaching system with high-speed video capture. *Journal of Neuroscience Methods*, 271:119–127, 2016. 3

[15] K. K. Fenrich, Z. May, A. Torres-Espín, J. Forero, D. J. Bennett, and K. Fouad. Single pellet grasping following cervical spinal cord injury in adult rat using an automated full-time training robot. *Behavioural brain research*, 299:59–71, 2016. 3

[16] K. Gale, H. Kerasidis, and J. R. Wrathall. Spinal cord contusion in the rat: behavioral analysis of functional neurologic impairment. *Experimental neurology*, 88(1):123–134, 1985. 2

[17] H. Grabner, M. Grabner, and H. Bischof. Real-time tracking via on-line boosting. In *BMVC*, volume 1, page 6, 2006. 2

[18] J.-Z. Guo, A. R. Graves, W. W. Guo, J. Zheng, A. Lee, J. Rodríguez-González, N. Li, J. J. Macklin, J. W. Phillips, B. D. Mensh, et al. Cortex commands the performance of skilled movement. *eLife*, 4:e10774, 2015. 3

[19] B. Hariharan, J. Malik, and D. Ramanan. *Discriminative Decorrelation for Clustering and Classification*, pages 459–472. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012. 7, 8

[20] S. Hong, T. You, S. Kwak, and B. Han. Online tracking by learning discriminative saliency map with convolutional neural network. *arXiv preprint arXiv:1502.06796*, 2015. 2

[21] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014. 5

[22] S. Johnson and M. Everingham. Clustered pose and nonlinear appearance models for human pose estimation. In *Proceedings of the British Machine Vision Conference*, 2010. doi:10.5244/C.24.12. 2, 7, 8

[23] M. Kabra, A. A. Robie, M. Rivera-Alba, S. Branson, and K. Branson. Jaaba: interactive machine learning for automatic annotation of animal behavior. *nature methods*, 10(1):64–67, 2013. 1, 3, 7

[24] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012. 3, 5, 7, 8

[25] H. Li, Y. Li, and F. Porikli. Deeptrack: Learning discriminative feature representations online for robust visual tracking. *IEEE Transactions on Image Processing*, 25(4):1834–1848, 2016. 2

[26] P. Liang, Y. Pang, C. Liao, X. Mei, and H. Ling. Adaptive objectness for object tracking. 2015. 2

[27] N. T. Lindau, B. J. Bänninger, M. Gullo, N. A. Good, L. C. Bachmann, M. L. Starkey, and M. E. Schwab. Rewiring of the corticospinal tract in the adult rat after unilateral stroke and anti-nogo-a therapy. *Brain*, 137(3):739–756, 2014. 2, 5

[28] C. Liu. *Beyond pixels: exploring new representations and applications for motion analysis*. PhD thesis, Citeseer, 2009. 3, 5

[29] T. Malisiewicz, A. Gupta, and A. A. Efros. Ensemble of exemplar-svms for object detection and beyond. In *ICCV*, 2011. 7, 8

[30] I. Misra, C. L. Zitnick, and M. Hebert. Unsupervised learning using sequential verification for action recognition. *arXiv preprint arXiv:1603.08561*, 2016. 2

[31] R. Morris and I. Q. Whishaw. A proposal for a rat model of spinal cord injury featuring the rubrospinal tract and its contributions to locomotion and skilled hand movement. *Frontiers in neuroscience*, 10, 2016. 2

[32] J. C. Niebles, C.-W. Chen, and L. Fei-Fei. *Modeling Temporal Structure of Decomposable Motion Segments for Activity Classification*, pages 392–405. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010. 2, 7

[33] M. Noroozi and P. Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. *arXiv preprint arXiv:1603.09246*, 2016. 2

[34] L. Pishchulin, E. Insafutdinov, S. Tang, B. Andres, M. Andriluka, P. V. Gehler, and B. Schiele. Deepcut: Joint subset partition and labeling for multi person pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4929–4937, 2016. 2, 8

[35] V. Ramakrishna, D. Munoz, M. Hebert, A. J. Bagnell, and Y. Sheikh. Pose machines: Articulated pose estimation via inference machines. In *ECCV*, 2014. 2, 8

[36] M. Ranzato, A. Szlam, J. Bruna, M. Mathieu, R. Collobert, and S. Chopra. Video (language) modeling: a baseline for generative models of natural videos. *arXiv preprint arXiv:1412.6604*, 2014. 2

[37] E. Shelhamer, J. Long, and T. Darrell. Fully convolutional networks for semantic segmentation. 2016. 2

[38] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. *arXiv*, abs/1406.2199, 2014. 2

[39] A. Toshev and C. Szegedy. Deeppose: Human pose estimation via deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1653–1660, 2014. 2

[40] C. Vondrick, H. Pirsiavash, and A. Torralba. Anticipating the future by watching unlabeled video. *arXiv preprint arXiv:1504.08023*, 2015. 2

[41] L. Wang, W. Ouyang, X. Wang, and H. Lu. Visual tracking with fully convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3119–3127, 2015. 2

[42] J. Wright, A. Ganesh, S. Rao, Y. Peng, and Y. Ma. Robust principal component analysis: Exact recovery of corrupted low-rank matrices via convex optimization. In Y. Bengio, D. Schuurmans, J. D. Lafferty, C. K. I. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems 22*, pages 2080–2088. Curran Associates, Inc., 2009. 3