VOLUME 17, N° 3 2023
Journal of Automation, Mobile Robotics and Intelligent Systems

Skeleton-based Human Action/Interaction Classification in Sparse Image Sequences

Submitted: 14th February 2023; accepted: 16th June 2023

Włodzimierz Kasprzak, Paweł Piwowarski

DOI: 10.14313/JAMRIS/3-2023/18

Abstract:
Research results on human activity classification in video are described, based on initial human skeleton estimation in selected video frames. Simple, homogeneous activities, limited to single person actions and two-person interactions, are considered. The initial skeleton data is estimated in selected video frames by software tools, like "OpenPose" or "HRNet". Main contributions of presented work are the steps of "skeleton tracking and correcting" and "relational feature extraction". It is shown that this feature engineering step significantly increases the classification accuracy compared to the case of raw skeleton data processing. Regarding the final neural network encoder-classifier, two different architectures are designed and evaluated. The first solution is a lightweight multilayer perceptron (MLP) network, implementing the idea of a "mixture of pose experts". Several pose classifiers (experts) are trained on different time periods (snapshots) of visual actions/interactions, while the final classification is a time-related pooling of weighted expert classifications. All pose experts share a common deep encoding network. The second (middle weight) solution is based on a "long short-term memory" (LSTM) network. Both solutions are trained and tested on the well-known NTU RGB+D dataset, although only 2D data are used. Our results show comparable performance with some of the best reported LSTM-, Graph Convolutional Network-(GCN), and Convolutional Neural Network-based classifiers for this dataset. We conclude that, by reducing the noise of skeleton data, highly successful lightweight- and midweight-models for the recognition of brief activities in image sequences can be achieved.

Keywords: *Action classification, Skeleton features, 2-person interactions, Mixture of experts, Video analysis*

1. Introduction

Human activity recognition has recently caught the attention of the computer vision community since it drives real-world applications that make our life better and safer, such as human-computer interaction in robotics and gaming, video surveillance, and social activity recognition [1]. For example, new robotic applications try to predict human activity patterns in order to let the robot early infer when a specific collaborative operation will be requested by the human [2, 3]. In video surveillance, human activity classification can be integrated with probabilistic prediction models, in order to infer the ongoing activity [4]. Ambient assited living technologies allow the recognition of a human's daily living activity in order to take care of dependent people [5].

The computer vision approach to human activity recognition in video clips or video streams is typically understood as the detection and classification of brief, homogeneous single-person actions and two-person interactions. A longer video may contain various actions that eventually are parts of more complex human activities. An action or interaction is decomposed in time into human poses, being recognized in single frames.

In early solutions, hand-designed features like edges, contours, Scale-Invariant Feature Transform (SIFT), and Histogram of Oriented Gradients (HOG) have usually been used for detection and localization of human body parts or key points in the image [6]. More recently, neural network-based solutions were successfully proposed, e.g., based on Deep Neural Networks (DNN) [7], especially Convolutional Neural Networks (CNN), LSTMs, and Graph CNNs [8], as they have the capability automatically to learn rich semantic and discriminative features. Furthermore, DNNs have an ability to learn both spatial and temporal information from signals and can effectively model scale-invariant features as well.

The approaches to vision-based human activity recognition can be divided into two main categories: activity recognition directly in video data [9] or skeleton-based methods [10], where the 2D or 3D human skeletons are detected first, sometimes already by specialized devices, like the Microsoft Kinect. The skeleton-based methods compensate some of the drawbacks of vision-based methods, such as assuring the privacy of participants and reducing the scene's light sensitivity. Some popular solutions to human pose estimation (i.e., the detection and localization of humans in images) can be mentioned: OpenPose [11], DeepPose [12], and DeeperCut [13]. It must be noticed, that the term "pose estimation" is commonly used in image and video analysis literature to refer to a result of semantic image segmentation models. Such models are trained to detect and classify objects, and to estimate by regression methods the image locations of object parts.

Although the human activity classification domain has gained noticeable improvements in recent years, it has still been facing many challenges in practice, e.g. occlusions, low resolutions, different view-points, non-rigid deformations, and intra-class variability in shape [7]. In this work, we deal with analysis of brief video clips or streams, that contain single-person actions and two-person interactions, assuming the existence of human skeleton data for selected video frames. After analysing the recent alternatives [14–16], we identified their main trends: (1) a smart processing of skeleton data for extracting meaningful information and cancelation of noisy data (e.g., relational networks); (2) designing light-weight and middle-weight solutions instead of heavy-weight networks by employing background knowledge, e.g., using graph CNNs instead of CNNs and LSTMs, or CNNs with 2-D kernels instead of 3-D CNNs.

The aim of our work was to create an efficient and practical tool for brief human action and interaction recognition in video data. Thus, we decided to propose light-weight deep network models, which operate on strong relational features (extracted in an application-aware skeleton processing step). For a solid verification of our contribution, a fair comparison with other known solutions, we decided to train and test the proposed solutions on a well known annonated video dataset (the NTU RGB+D).

For the two considered problems (action- and interaction classification), we propose two solutions – one lightweight model, based on feedforward MLPs, and one middle-weight model, based on LSTM network. In first case, we took the idea of extending pose classification in still images to activity classification in video by applying a new version of the well-known pattern classification approach, called "mixture of experts" [17]. But instead of identifying subdomains or learning different weights for expert classifiers in subdomains, the experts are distributed along the time axis. Every expert is responsible for classification of frames belonging to different time periods of an activity process, e.g., start, initial, midterm, final or end period. The final classification is obtained by a weighted evaluation of class likelihoods of all the expert pose classifiers. With the second proposed solution, an LSTM model, and with the application-aware feature engineering step [18], we try to compete with the best performing methods in this field, which use Graph CNNs or 3D CNNs.

There are four remaining sections of this work. Section 2 describes recent approaches in human action and -interaction recognition. Our solutions, the feature engineering step and the two deep network models – one based on ANN pose experts and one on an LSTM network – are introduced in Section 3. Next, in Section 4, results of experiments are described and conclusions are drawn. Finally, in Section 5, we summarize our contribution to the subject.

2. Related Work

Typically, human activity recognition in images and video requires first a detection of human body parts or key-points of a human skeleton. The skeleton-based methods compensate some of the drawbacks of vision-based methods, such as assuring the privacy of persons and reducing the scene's light sensitivity. They also limit the sensitivity to clothes, hair, and other person-specific features.

2.1. Human Pose Estimation

Typically, a human's pose is represented by the localisation of image features, key points or body parts, expressed in camera coordinates. In the past, mainly hand-crafted features have been used, such as edges, contours, Scale-Invariant Feature Transform (SIFT) or Histogram of Oriented Gradients (HOG). However, these approaches have produced modest performance when it comes to accurately localizing human body parts [19]. With the development of Convolutional Neural Networks (CNNs), the performance in solving human pose estimation problems has improved constantly and been significantly higher than the traditional methods [14].

There are three fundamental architectures, AlexNet [20], VGG [21], and ResNet [22], which have been employed as the backbone architecture for many human pose estimation studies [23]. Released in 2012, AlexNet has been considered one of the backbone architectures for many computer vision models. The DeepPose software employed AlexNet for estimating human poses [12]. Popular works in pose estimation, OpenPose [11], and human parts detection, Faster RCNN [24], have used VGG and achieved state-of-the-art performance in visual human estimation. After the release of ResNet, many works on human pose estimation have applied it as a backbone (e.g., [13, 25]).

2.2. Human Action and Interaction Recognition

In the machine learning literature it is often said that the skeleton data consists of "joints" and "limbs" (or "bones"). We must admit, that the terms "joints" and "bones" have no direct correspondence to human body joints and bones. In order to clarify the terminology, we should rather call them "key points of the skeleton" and "skeletal segments", accordingly.

The vast majority of research on human action and interaction recognition is based on the use of artificial neural networks. However, initially, classical approaches have also been tried, such as the SVM (e.g. [26, 27]). Yan et al. [28] used multiple features, like a "bag of interest points" and a "histogram of interest point locations", to represent human actions. They proposed a combination of classifiers in which AdaBoost and sparse representation (SR) are used as basic algorithms. In the work of Vemulapalli et al. [29], human actions are modeled as curves in a Lie group of Euclidean distances. The classification process is a combination of dynamic time warping, Fourier temporal pyramid representation, and linear SVM.

Thanks to higher quality results, artificial neural networks are replacing other methods. Thus, the most recently conducted research in the area of human activity classification differs only by the proposed network architecture. Networks based on the LSTM architecture or a modification of this architecture (a ST-LSTM network with trust gates) were proposed by Liu et al. [30] and Shahroudy et al. [31]. They introduced so called "Trust Gates" for controlling the content of an LSTM cell and designed an LSTM network capable of capturing spatial and temporal dependencies at the same time (denoted as ST-LSTM). The task performed by the gates is to assess the reliability of the obtained joint positions based on the temporal and spatial context. This context is based on the position of the examined junction in the previous moment (temporal context) and the position of the previously studied junction in the present moment (spatial context). This behavior is intended to help network memory cells assess which locations should not be remembered and which ones should be kept in memory. The authors also drew attention to the importance of capturing default spatial dependencies already in the skeleton data. They have experimented with different mappings of the a joint's set to a sequence. Among the, they mapped the skeleton data into a tree representation, duplicating joints when necessary to keep spatial neighborhood relation, and performed a tree traversal to get a sequence of joints. Such an enhancement of the input data allowed an increase of the classification accuracy by several percent.

The work [32] introduced the idea of applying convolutional filters to pseudo-images in the context of action classification. A pseudo-image is a map (a 2D matrix) of feature vectors from successive time points, aligned along the time axis. Thanks to these two dimensions, the convolutional filters find local relationships of a combined time-space nature. Liang et al. [33] extended this idea to a multi-stream network with three stages. They use 3 types of features, extracted from the skeleton data: positions of joints, motions of joints and orientations of line segments between joints. Every feature type is processed independently in its own stream but after every stage the results are exchanged between streams.

Graph convolutional networks are currently considered a natural approach to the action (and interaction) recognition problem. They are able to achieve high quality results with only modest requirements of computational resources. "Spatial Temporal Graph Convolutional Networks" [34] and "Actional-Structural Graph Convolutional Networks" [35] are examples of such solutions.

Another recent development is the pre-processing of the skeleton data in order to extract different type of information (e.g., information on joints and bones, and their relations in space and time). Such data streams are first separately processed by so called multi-stream neural networks and later fused to a final result. Examples of such solutions are the "Two-Stream Adaptive Graph Convolutional Network" (2S-AGCN) and the "Multistream Adaptive Graph Convolutional Network" (MAGCN), proposed by Shi et al. [36, 37].

One of the best performances on the NTU RGB+D interaction dataset is reported in the work of Perez et al. [15]. Its main contribution is a powerful two-stream network with three-stages, called "Interaction Relational Network" (IRN). The network inputs are basic relations between joints of two interacting persons tracked over the length of image sequence. An important step is the initial extraction of relations between pairs of joints – both distances between joints and their motion are obtained. The neural network makes further encoding and decoding of these relations and a final classification. The first stream means the processing of within-a-person relations, while the second one – between-person relations. The use of a final LSTM with 256 units is a high-quality version of the IRN network, called IRN-LSTM. It allows to reason over the interactions during the whole video sequence – even all frames of the video clip are expected to be processed. In the basic IRN, a simple densely-connected classifier is used instead of the LSTM and a sparse sequence of frames is processed.

The currently best results for small networks are reported by Zhu et al. [16], where two new modules are proposed for a baseline 2S-AGCN network. The first module extends the idea of modelling relational links between two skeletons by a spatio-temporal graph to a "Relational Adjacency Matrix (RAM)". The second novelty is a processing module, called "Dyadic Relational Graph Convolution Block", which combines the RAM with spatial graph convolution and temporal convolution to generate new spatial-temporal features.

Very recently, exceptionally high performance was reported when using networks with 3D convolutional layers, applied to data sensors that constitute skeleton "heatmaps" (i.e., preprocessed image data) [38]. The approach, called PoseConv3D, can even be topped, when fused with the processing of ordinary RGB-data stream [39]. Obviously, this requires to create a heavy network and produces high computational load.

2.3. Conclusion

From the analysis of the recent most successful solutions, we can draw three main conclusions:

1) using an analytic preprocessing of skeleton-data to extract meaningful information and cancel noisy data, either by employing classic functions or learnable function approximations (e.g., relational networks);

2) preferring light-weight solutions by employing background (problem-specific) knowledge, i.e., using graph CNNs instead of CNN or CNNs with 2-D kernels instead of 3-D CNN;

3) a video clip containing a specific human action or interaction can be processed alternatively as a sparse or dense frame sequence, where sparse sequence is chosen to achieve real-time processing under limited computational resources, while the processing of a dense sequence leads to better performance.

3. The Approach

3.1. Structure

The input data has the form of video clips, containing a single-person action or a two-person interaction. The two proposed solutions have a common first part and differ by the feature extraction step and neural network-based classifier, as shown in Figure 1. The common processing steps are: key frame selection, skeleton estimation, skeleton tracking and -correcting. The final steps, customized for action and interaction recognition, are: feature extraction and deep neural network model.

Key frame selection The start- and end frames of an activity in a video clip are detected first. The idea of this detection follows a typical approach to voice activity detection in speech signal, but a hysteresis of thresholds for motion "energy" instead of signal energy is applied. The detected activity window will be represented by a fixed number N of video frames (e.g. $N = 32$) from M subintervals (groups). Thus, $N = M \cdot m$, where M is the number of consecutive subintervals (groups), while m is the number of frames in one group. In particular, let us fix $M = 4$ and denote the groups as follows: start, 1-st intermediate, 2-nd intermediate and final. For every group a separate *expert* (a weak action classifier) will be created.

Skeleton estimation In our implementation, we use the core block of *OpenPose* [44], the "body_25" model, to extract 25 human skeletal keypoints from an image. The result of OpenPose, as applied to a single key frame, will be a set of skeletons, where a 25-elementary array represents a single skeleton, providing 2D (or 3D, if needed) image coordinates and a confidence score for every keypoint (called "joint"). OpenPose provides the ability to obtain 2D or 3D information about detected joints. If one selects the 3D option, views must be specified for the library to perform triangulation. The library returns for every joint the following data:

- (x, y, p)—for 2D joints;
- (x, y, z, p)—for 3D joints.

where (x, y) are the image coordinates of a pixel, z is the depth associated with the given pixel, p is the certainty of detection and is a number in the range <0:1>. We have used the pretrained "OpenPose" model with default parameter settings and no person number limit. The received skeleton data was initially refined by deleting keypoints with certainty p below the certainty threshold of 0.1. Later, during skeleton refinement, such missing joints and remaining weak joints (with $p < 0.3$) will be approximated from the data of their well-detected neighbors (with $p \geq 0.3$).

The OpenPose library offers the possibility to choose a model of a human figure. There are three models: "15 body", "18 body", and "25 body". The number in the name refers to the number of detectable joints. Table 1 lists the typically selected model of the "25 body".

Table 1. List of keypoints (joints) in the "25 body" model

Number	Description
0	The main point of the head
1	Neck base
2, 5	Shoulders
3, 6	Elbows
4, 7	Wrists
8	The base of the spine
9, 12	Hips
10, 13	Knee
11, 14	Cube
15, 16, 17, 18	Extra head points
19, 20, 21, 22, 23, 24	Extra foot points

Skeleton tracking and correcting In case, more than two skeletons in an image are returned by OpenPose, the two largest skeletons, S_a, S_b, are selected first and then tracked in the remaining frames. We focus on the first 15 joints of every skeleton, as the information about the remaining joints is very noisy (Fig. 2).

In the case of many-person scenes, the sets of skeletons generated by OpenPose, are not uniquely indexed over the frame sequence. There may also be falsely detected skeletons for objects in the background, or a large mirror can lead to a second skeleton of the same person.

The algorithm for tracking skeletons in many-skeleton frames can be seen as a multiple path search and can be solved in many ways. For example, beam search or some type of dynamic programming search could be used. Our algorithm initializes paths for up to two "foreground" skeletons, detected in the first frame, and then runs a loop over the remaining frames trying to extend every path by the nearest skeleton detected in the next frame that is sufficiently close to the path's last skeleton. New paths can also start in later frames when apparently new persons appear on the scene.

The invariance of features with respect to the size of the skeleton in the image is obtained by normalizing the coordinates of the junction points with the section length between the neck $joint_1$ and the center of the hips $joint_8$.

The selected sequence of skeletons, representing one person, is sometimes deteriorated by missing joints of some skeleton or by missing an entire skeleton in a frame. These misses of joints or virtual replications of skeletons introduce noise and may also destabilize the feature normalization step when the locations of required joints are missing.

Let $\boldsymbol{v}_i$ be a series of N positions of the „i-th" joint in time: $v_i = [o_1^1, o_i^2, ..., o_i^N]$. The procedure for the refinement of joints can be summarized as follows:

1) IF some position o_i^t is missing THEN take the average of neighbors o_i^{t-1}, o_i^{t+1} in time;
2) in the middle of the frame sequence, a missing joint is set to a linear interpolation of the two closest-time known positions of this joint: IF $o_i^{(.)}$ is missing k consecutive times, i.e., from t to $t + k - 1$; THEN take interpolations of values $o_i^{(t-1)}, o_i^{(t+k)}$;

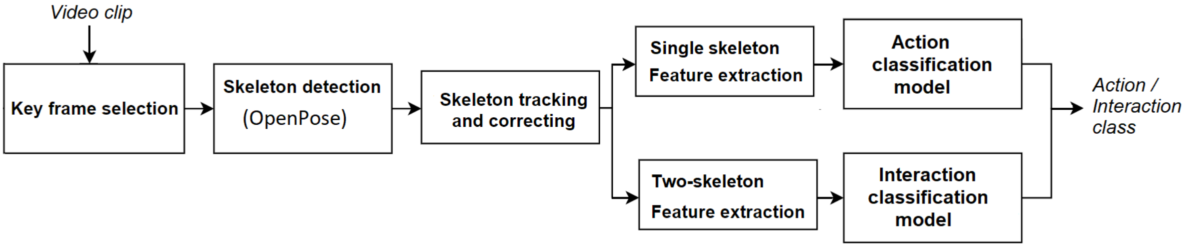

Video clip
Key frame selection
Skeleton detection
(OpenPose)
Skeleton tracking and correcting
Single skeleton Feature extraction
Action classification model
Two-skeleton Feature extraction
Interaction classification model
Action / Interaction class

Figure 1. General structure of our approach

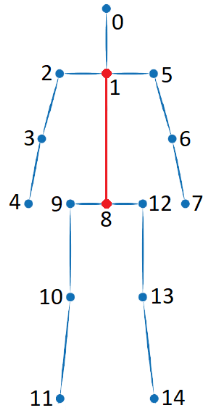


Figure 2. The 15 reliable keypoints (joints) (indexed from 0 to 14) out of 25 of the OpenPose's "body_25" skeleton model

3) for the initial frame, the position of a missing joint is set to its first detection in the frame sequence: IF $o_i^{(.)}$ is missing first k times; THEN set first k values to $o_i^t = o_i^{(k+1)}, (t = 1, ..., k)$;
4) joints that are lost at the end of the frame sequence receive the positions last seen: IF $o_i^{(.)}$ is missing last k times; THEN set last k values to $o_i^t = o_i^{(N-k)}, (t = N - k + 1, ..., N)$;
5) IF $o_i^{(.)}$ is completely missing in the entire sequence; THEN set the joint data according to its sister joint, i.e., obtain a symmetric mapping (w.r.t. the spin axis) of "visible" sister joint.

3.2. Feature Extraction

The result of tracking (up to) two sets of skeleton joints in N frames can be represented as a 2D map of $N \times 15 \times 2$ vector entries, where 2 means image coordinates (x, y). We call this structure as **RAW** features.

A representation of junction positions (the RAW features) has the disadvantage of being not invariant to basic transformations of the image space. It does not explicitly represent relationships within a skeleton and between two skeletons, like motion of joints and relative orientation of branches. Therefore, a relational representation of both skeletons was developed, which reduces the aforementioned disadvantages of the raw representation of joints. The new features consist of:

1) int-PS (”polar sparse” for interaction) features: with a local center S and normalization vector **u**, defined for a pair of skeletons, vectors are drawn between point S and every joint of the two skeletons – all these vectors are length- and orientation-normalized w.r.t. $\boldsymbol{u}$;

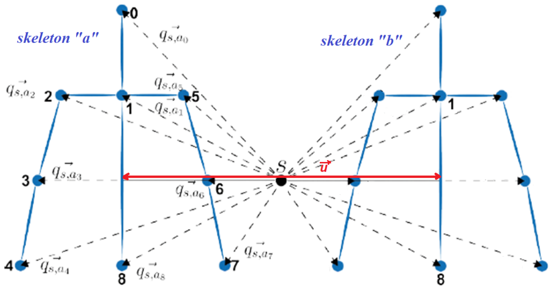


Figure 3. Illustration of the PS features: the vectors between reference point S and every skeleton joint, normalized by vector $\boldsymbol{u}$

2) int-PSM (”polar sparse with motion” for interaction) features: the int-PS features with additional motion vectors of all the joints.

For memory-less networks, like the MLP, in order to make them competitive to LSTM models, we add motion vectors for every joint of every skeleton. The int-PS features will be fed into the LSTM-based classifier, while the int-PSM features – into the ensemble of pose-based classifiers (the "mixture of experts" model).

int-PS Let us define the center point S of vector $\boldsymbol{u}$, which connects the central spin points of both skeletons (Fig. 3). Now 15 vectors are defined for every skeleton. Every vector connects the point S with a joint of skeleton 1 or 2. Every vector is represented in polar form by two features – normalized magnitude $h_{a,j}, h_{b,j}$ and relative orientation $r_{a,j}, r_{b,j}$ (both magnitude and orientation are normalized with respect to magnitude and orientation of $\boldsymbol{u}$). Thus, for every frame there are 60 features defined (= $(15 + 15) \cdot 2$). The $N \cdot 60$ features are split into two maps, H_a^N and H_b^N, one for each skeleton:

$$H_a^N = \begin{bmatrix} \boldsymbol{h}_a^1 & \smile & \boldsymbol{r}_a^1 \\ \boldsymbol{h}_a^2 & \smile & \boldsymbol{r}_a^2 \\ ... & \smile & ... \\ \boldsymbol{h}_a^N & \smile & \boldsymbol{r}_a^N \end{bmatrix} \tag{1}$$

$$H_b^N = \begin{bmatrix} \boldsymbol{h}_b^1 & \smile & \boldsymbol{r}_b^1 \\ \boldsymbol{h}_b^2 & \smile & \boldsymbol{r}_b^2 \\ ... & \smile & ... \\ \boldsymbol{h}_b^N & \smile & \boldsymbol{r}_b^N \end{bmatrix} \tag{2}$$

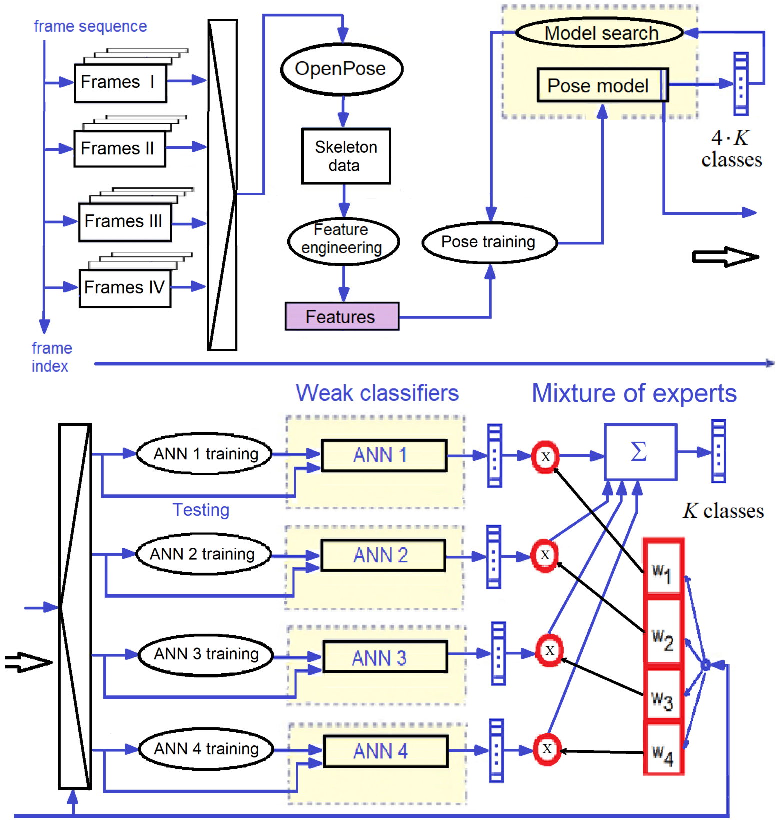


Figure 4. Structure of the "mixture-of-pose-experts" model

act-PS For single-person action classification, one feature map H_a^N, with 30 features, is created only. In this case, the reference point S is the center point of spin segment $o_1 \frown o_8$ and the vector $\mathbf{u}$ is the spin segment $o_1 \frown o_8$.

int-PSM In addition to the int-PS features, motion vectors $(\delta x_{a,j}^i, \delta y_{a,j}^i)$ are defined for every joint of the two skeletons. Thus, $((15+15) \times 2) = 60$ features are added and the int-PSM vector consists of 120 features.

act-PSM For single-person action classification, the act-PS feature vector is extended by motion vectors of the single skeleton joints only. Thus, 30 features are added, and the act-PSM vector has 60 features in total.

3.3. Mixture of Pose Experts (MPE)

The first neural network model is called "mixture of pose experts" (shortly: MPE) (Fig. 4). We can distinguish three parts of this network: one pose encoder/classifier, four pose-based activity classifiers ("pose experts"), and a final activity classifier (fusing the results of pose experts).

There is a common pose encoding network, which is trained with $4 \times K$ pose classes, i.e., every pose class represents one of the K actions (or interactions) in one of the four time periods. After the training is accomplished, the classification layers are omitted, and the encoding embedding vector is passed to the four weak pose classifiers (called "experts"). Thus, the output layer of the pose classifier is replaced by every expert network – a fully connected hidden layer with K outputs each. Four alternative classifiers are trained on the PSM features obtained for samples from the time period corresponding to the given pose classifier.

The mixture of experts network (MPE) is implemented using Keras [45]. During training, the models are evaluated, and a search for optimal model parameters is performed – a *RandomSearch* algorithm from Keras is applied in this stage. The following options of the MLP parameters are evaluated: the number of hidden layers of the network can vary from 1 to 3, different activation functions (ReLU and/or sigmoid) may be chosen, as well as the number of neurons in hidden layers and the learning rate can vary. The ensemble classifier consists of a fusion of expert results and an aggregation of class likelihoods over the entire frame sequence. The fusion layer is again a fully connected layer that is weighting the results of all weak classifiers. It takes the frame index t as it is an additional input.

Activity (i.e., action or interaction) classification then consists of specifying likelihoods of all activity classes by aggregating their corresponding pose likelihoods over the entire time sequence. The aggregation operation is mathematically a weighted sum of pose likelihoods for frames indexed from $t = 1$ to $t = N$. The ensemble classifier provides gain coefficients for the four experts depending on the frame index:

$$\boldsymbol{S} = \sum_{t=1}^{N} [\boldsymbol{Pr}_{expert\_1}(t) \cdot w_1(t) + \boldsymbol{Pr}_{expert\_2}(t) \cdot w_2(t) + \boldsymbol{Pr}_{expert\_3}(t) \cdot w_3(t) + \boldsymbol{Pr}_{expert\_4}(t) \cdot w_4(t) \tag{3}$$

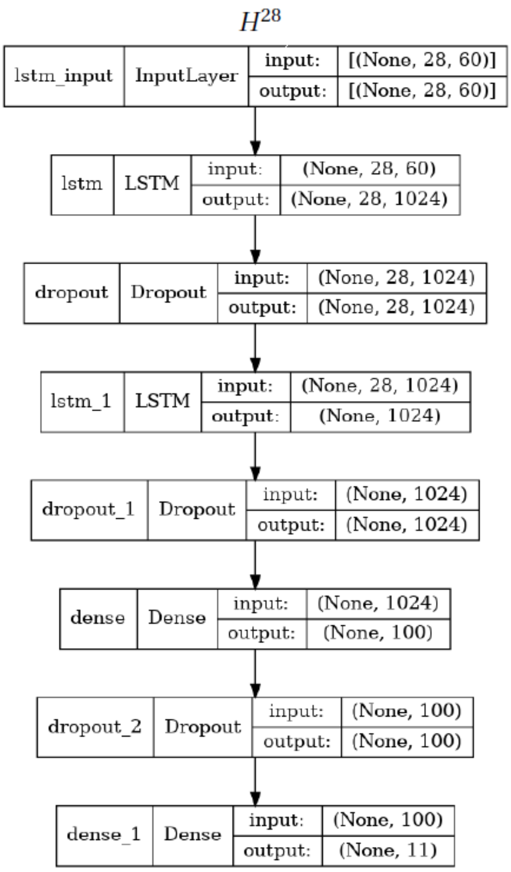

H^{28}
lstm_input | InputLayer | input: [(None, 28, 60)] | output: [(None, 28, 60)]
lstm | LSTM | input: (None, 28, 60) | output: (None, 28, 1024)
dropout | Dropout | input: (None, 28, 1024) | output: (None, 28, 1024)
lstm_1 | LSTM | input: (None, 28, 1024) | output: (None, 1024)
dropout_1 | Dropout | input: (None, 1024) | output: (None, 1024)
dense | Dense | input: (None, 1024) | output: (None, 100)
dropout_2 | Dropout | input: (None, 100) | output: (None, 100)
dense_1 | Dense | input: (None, 100) | output: (None, 11)

Figure 5. Architecture of the SC-LSTM-PS network for $N = 28$ key frames. The versions of SC-LSTM differ only by the input layer size

3.4. LSTM Model

A "single channel" LSTM network (denoted further as SC-LSTM) is proposed. It consists of two LSTM layers, interleaved by two Dropout layers, and two dense layers (Fig. 5). Depending on the role of given network model, we can distinguish between the action- and interaction-classification models (SC-LSTM-act, SC-LSTM-int). In turn, based on the type of input features (RAW or PS/PSM) every such model will appear in two versions (e.g., SC-LSTM-act-RAW, SC-LSTM-int-PS). Recall that the two baseline feature versions SC-LSTM-act-RAW and SC-LSTM-int-RAW process the raw skeleton joints, as initially obtained by OpenPose and established by the skeleton tracking and refinement step.

These versions differ by the input layer only, as there are different numbers of features considered. For example, in the SC-LSTM-act-PS version, there are 3,350 k trainable parameters.

4. Results

4.1. Datasets

Action set The dataset "NTU RGB+D" [31] is the basic set used in this work. It was made by ROSE (Rapid-Rich Object Search Lab), which is the result of a collaboration between Nanyang Technological University in Singapore and Peking University in China. Many works on human action recognition have already been validated on this dataset, and a website collects the achieved performance scores [14]. The set can be characterized as follows (Fig. 6):

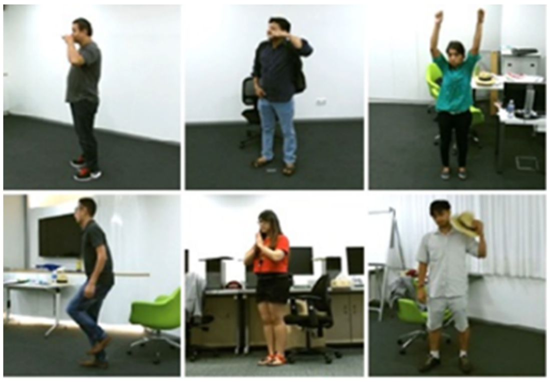


Figure 6. Image samples of actions from the NTU RGB+D action dataset (from left to right and top to bottom): drink, eat, hand waving, jump up, put palms together, take off a hat

Table 2. The "everyday activities" subset (40 classes) of the NTU-RGB+D dataset

A1: drink water	A2: eat meal
A3: brush teeth	A4: brush hair
A5: drop	A6: pick up
A7: throw	A8: sit down
A9: stand up	A10: clapping
A11: reading	A12: writing
A13: tear up paper	A14: put on jacket
A15: take off jacket	A16: put on a shoe
A17: take off a shoe	A18: put on glasses
A19: take off glasses	A20: put on a hat/cap
A21: take off a hat/cap	A22: cheer up
A23: hand waving	A24: kicking something
A25: reach into pocket	A26: hopping
A27: jump up	A28: phone call
A29: play with phone/tablet	A30: type on a keyboard
A31: point to something	A32: taking a selfie
A33: check time (from watch)	A34: rub two hands
A35: nod head/bow	A36: shake head
A37: wipe face	A38: salute
A39: put palms together	A40: cross hands in front

- Contains RGB videos with a resolution of 1920 × 1080 (pixels).
- Includes depth and infrared maps with a resolution of 512 × 424 (pixels).
- Each behavior of the set is captured by three cameras.
- Behaviors were performed by people in two settings (showing activities from different viewpoints).
- It consists of 56,880 videos showing 60 classes of behavior.

Among the classes of behavior, the most popular are the "everyday activities". They constitute a subset of 40 classes, as listed in Table 2. The collection contains 37,920 video clips, and associated depth maps and infrared frames.

Table 3. The 10 activity classes of the UTKinect-Action3D dataset

A1: walk	A2: sit down
A3: stand up	A4: pick up
A5: carry	A6: throw
A7: push	A8: pull
A9: wave hands	A10: clap hands

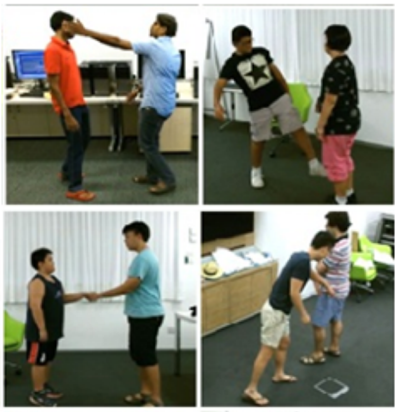


Figure 7. Image samples from the NTU RGB+D interaction dataset – interaction classes "Punch/slap, kicking, shaking hands, touch pocket"

The NTU RGB+D dataset allows to perform a cross-subject (person) (short: CS) or a cross-view (CV) evaluation. In the cross-subject setting, samples used for training show actions performed by half of the actors, while test samples show actions of remaining actors. In the cross-view setting, samples recorded by two cameras are used for training, while samples recorded by the remaining camera – for testing. A major advantage of this dataset is an exact specification which video clips should be used for training and which for testing.

The UTKinect-Action3D dataset [46] is the second set of people's activities used in this work. This set will be a secondary set, which means that only a testing of the developed model will be performed on it. The UTKinect dataset can be described by the following:

- Includes RGB videos with 640 × 480 resolution (pixels).
- Includes depth maps with a resolution of 320 × 240 (pixels).
- Activities were performed by 10 people. Each person repeated the performed activity 2 times. There are 10 activity classes.

In Table 3, there are listed the 10 activity classes in UTKinect-Action3D.

The clips in this collection are organised as photo series in catalogs. To view a specific video, images within each catalog must be collected. A single video contains a person performing a series of 10 actions. A video is labeled with the action class, start photo and end photo of every action.

Interaction set The best configuration of the pose experts and the final, time-accumulating network will be trained and tested on the interaction subset of the NTU RGB+D dataset (Fig. 7). It includes 11 two-person interactions of 40 actors: A50: punch/slap, A51: kicking, A52: pushing, A53: pat on back, A54: point finger, A55: hugging, A56: giving object, A57: touch pocket, A58: shaking hands, A59: walking towards, and A60: walking apart. In our experiments, the skeleton data of the NTU-RGB+D dataset is already considered. There are 10,347 video clips in total, in which 7,334 videos are in the training set and the remaining 3,013 videos are in the test set. No distinct validation subset is distinguished, as the idea is to run sufficient numbers of training/testing iterations and to select the best test iteration.

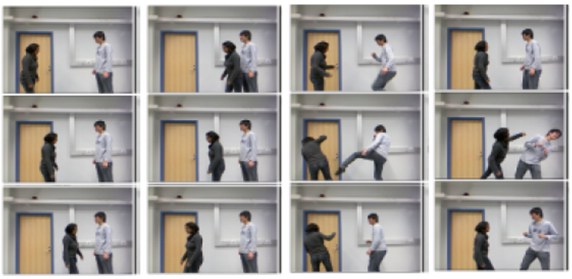


Figure 8. Image samples from the SBU Kinect interaction dataset – interaction classes (from left to right) "moving toward, moving apart, kicking, slapping"

Skeleton data may consist of 25 joints of 3D skeletons that apparently represent a single person. As our research objective is to analyse video data and focus on only reliably detected joints, we use only the 2D information of only the first 15 joints. From a video sample, a set of frames is chosen as follows: the video clip is uniformly split into $M = 4$ time intervals ("periods"), from every period some number of frames m is selected. We tested $m = 2, 4, 6, 8$ and found that $m = 8$, giving $N = 32$ is the best setting.

A second interaction dataset, used here mainly for initial parameter search, is the SBU Kinect Interaction dataset [47]. There are two person interactions of 8 types recorded by Microsoft Kinect v1: moving toward, moving apart, pushing, kicking, slapping, giving objects, hugging, and hand shaking (Fig. 8). There are 21 subsets of image sequences recorded with pairs of actors (7 actors in total), performing all 8 interactions – all in the same single environment. In total, there are around 300 video samples – images of resolution 640 × 480 – recorded with time rate of $15fps$.

4.2. Interaction Classification

Pose model search For running the *RandomSearch* algorithm, an *NNHyperModel* is created, which implements the *HyperModel* class from the Keras tuner. The hyper-parameters of the search space are declared in *NNHyperModel* as class parameters. We trained and tested various configurations of the Pose model and the mixture model on the SBU Interaction and UT-Action datasets. For a given number of hidden layers (1, 2 or 3), we searched for the best number of neurons, the best type of activation function and the best value of the learning rate. Using the *RandomSearch*

Table 4. The mean accuracy on the SBU Interaction dataset of three pose expert models (PE) with 1, 2 and 3 hidden layers

mAP	1	2	3
PE – training mAP	95%	96%	99%
PE – test mAP	82%	84%	82%

Table 5. The accuracy of pose experts (PE), and the accuracy of the mixtures of experts MPE, verified on the NTU RGB+D interaction dataset in the CS (cross subject) mode

Classifier	Training	Test
Pose expert	88.4%	76.8%
MPE	94.6%	84.0%

function during training, we identified three ANN configurations, each one best performing for given number of hidden layers (1, 2 or 3).

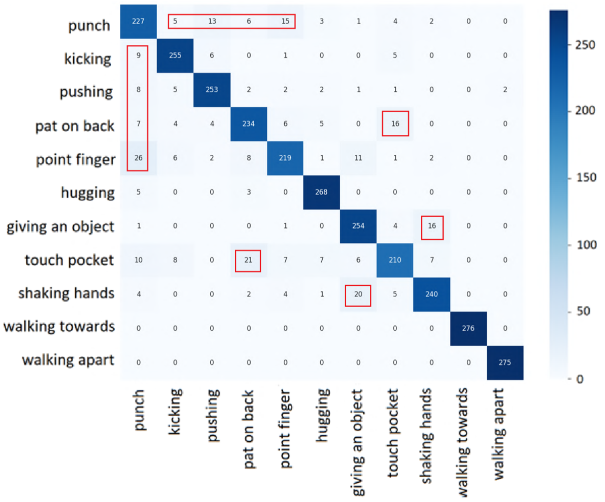
The performances of the three selected models after 100 epochs of training on the SBU Interaction dataset are shown in Table 4. The best mean test *accuracy* (i.e., the *recall* averaged over all classes) of 84% was achieved by the second model, whereas the other two have shown an accuracy of 82%. Consequently, we have chosen an ANN configuration of 2 hidden layers with 700 and 500 neurons in the first and second layer, respectively. The activation functions are ReLU and sigmoid, respectively.

Mixture of ANN experts The finally chosen ANN pose experts follow the second version, with two hidden layers, as reported above. The final score of every interaction class is obtained by the fusion network with final accumulation over time. The class with the highest score is selected as the winner. A notable improvement is observed, when fusing the results of experts by the ensemble classifier. The mean accuracy of a pose expert (PE) was 88.4% (training) and 76.8% (testing), while the ensemble classifier MPE has reached 94.6% and 84.0%, respectively (Table 5).

Processing times Experiments were conducted on a personal computer with processor Intel® Core™ i7-7700HQ CPU @ 2.80GHz, GPU Nvidia GeForce GTX 1060Mobile with Nvidia CUDA driver, 16 GB RAM DDR4 2400MHz and a PM961 NVMe SAMSUNG 256GB + HDDWestern Digital Blue 1TB. The average processing times are shown in Table 6.

Comparison with related methods Many methods for two-person interaction classification have been tested on the NTU RGB+D interaction dataset. We list some of the leading works in Table 7. The results can be characterized as follows:

- (a) the mixture of pose experts MPE-int needs a low number of weights (456 K) to be trained but achieves good quality (84%);
- (b) the single-channel LSTM-int needs a reasonable (midweight) number of weights (3.35 M) but achieves high quality (91.2%), slightly lower than current best approaches, based on graph CNNs [16] and 3D CNNs [39];
- (c) Solutions, that process all or nearly all frames of the video clip demonstrate superior performance over solutions operating on sparse frame sequences.



	punch	kicking	pushing	pat on back	point finger	hugging	giving an object	touch pocket	shaking hands	walking towards	walking apart
punch	227	5	13	6	15	3	1	4	2	0	0
kicking	9	255	6	0	1	0	0	5	0	0	0
pushing	8	5	253	2	2	2	1	1	0	0	2
pat on back	7	4	4	234	6	5	0	16	0	0	0
point finger	26	6	2	8	219	1	11	1	2	0	0
hugging	5	0	0	3	0	268	0	0	0	0	0
giving an object	1	0	0	0	1	0	254	4	16	0	0
touch pocket	10	8	0	21	7	7	6	210	7	0	0
shaking hands	4	0	0	2	4	1	20	5	240	0	0
walking towards	0	0	0	0	0	0	0	0	0	276	0
walking apart	0	0	0	0	0	0	0	0	0	0	275

Figure 9. Example of confusion matrix for NTU RGB+D interaction classes, obtained by the SC-LSTM model

Confusion matrix Confusion matrices allow for accurate analysis of incorrect predictions of individual classes. Figure 9 shows an example of a confusion matrix obtained for the NTU RGB+D interaction classes. The number of test samples has been virtually made equal for all the classes, thus the number of 276 positive results means a 100% accuracy for given class. We show results of an average performing model, so that mistakes are better visible than in cases of better performing models. It can be seen that the vast majority of class predictions are correctly done. As the dataset provides balanced sets for all classes, the simple accuracy measure is used: $Acc = (TP/All) \cdot 100\%$, where TP means “true positive results” and All – all test samples.

”Punch” (A50) is misclassified with A51-A54, which all use hands to express an action. In turn, the “finger pointing” class (A54) is mainly confused with the “punch” class (A50). In both cases, a similar hand movement is made towards the other person. The class of “pat on back” (A53) is confused with the class of “touch pocket” (A57). Both movements involve touching another person on their back. The “giving object” class (A56) and the “shaking hands” class (A58) represent very similar interactions – both involve the contact of the hand. The “walking towards” and “walking apart” classes are detected highly accurate.

4.3. Action Classification

Comparison with related methods Our two models, designed for single-person action classification (MPE-act-PSM, SC-LSTM-act-PS) are compared with other top-performing solutions, when trained and tested on

Table 6. Processing times measured for a video clip of 3 s

Step	Per frame	32 frames	Remarks
Activity detection	0.5 – 5 ms	16 – 160 ms	in 2 – 20 frames
Skeleton detection	67 ms	2134 ms	OpenPose
Feature engineering	8 ms	256 ms	
Pose classifier	1 ms	32 ms	
Mixture classifier	1 ms	32 ms	
Total time	78 – 87 ms	2470 – 2614 ms	

Table 7. Interaction classification accuracy of leading works evaluated on the NTU-RGB+D interaction set in the CS (cross subject) mode. Note: † – result according to [15], ‡ – result according to [16]

Model	Year	Accuracy (CS)	Parameters	Sequence
ST-LSTM [30]	2016	83.0% †	$\sim 2.1M$	32
ST-GCN [34]	2018	83.3% †	$3.08M$	32
AS-GCN [35]	2019	89.3% †	$\sim 9.5M$	32
IRN$_{inter+intra}$ [15]	2019	85.4% †	$\sim 9.0M$	32
MPE-int-RAW	2022	76.1%	$0.456M$	32
SC-LSTM-int-RAW	2022	80.2%	$3.35M$	32
MPE-int-PSM	2022	84.0%	$0.456M$	32
SC-LSTM-int-PS	2022	91.2%	$3.35M$	32
LSTM-IRN [15]	2019	90.5% †	$\sim 9.08M$	$\max(all, 300)$
2S-AGCN [36]	2019	93.4% ‡	$3.0M$	$\max(all, 300)$
DR-GCN [16]	2021	93.6% ‡	$3.18M$	$\max(all, 300)$
2S DR-AGCN [16]	2021	94.6% ‡	$3.57M$	$\max(all, 300)$
PoseConv3D(J+L) [39]	2022	97.0% ‡	$6.9M$	$\max(all, 300)$

the NTU RGB+D dataset (i.e., the everyday activity subset). The results in the CV (cross-view verification) mode are shown in Table 8. Typically, the accuracy obtained in CV mode used to be by a significant margin higher than in CS mode. But the number of action classes (40) is much higher than the number of interaction classes (11), previously considered.

There is a visible tradeoff between the complexity of a solution and classification accuracy – our lightweight MPE-act-PSM vs. complex solutions of three-stream CNNs (3S-CNN) or solutions using Graph CNNs and 3D convolutions of skeleton heatmaps (PoseC3D). Exceptionally high performance was recently reported, when fusing RGB data and skeleton data processing results [39]. However, the relatively early methods have performed worse than ours. Even the best among them, mentioned above as ST-LSTM, has reached a performance level barely comparable with our raw skeleton data (RAW) processing methods.

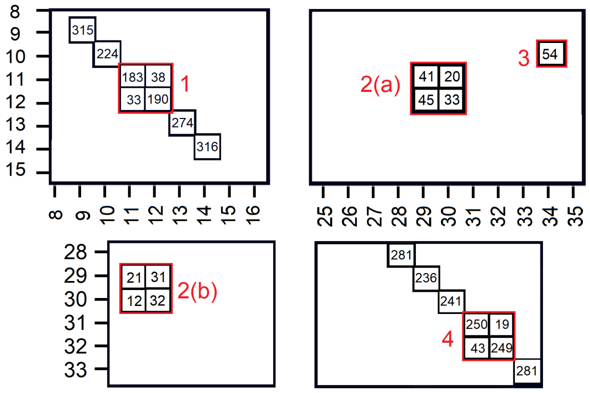


Figure 10. The main confusions between 40 classes of the NTU RGB+D action dataset: four main confusion cases (given by red numbers 1, 2, 3 and 4, where confusion no. 2 is a symmetric relation) in the 40 × 40 confusion matrix

Confusion matrix Figure 10 shows the main confusions detected for 40 classes of the NTU RGB+D dataset. The confused classes can be divided into three groups. Actions numbered 11 (read), 12 (write), 29 (play phone/tablet) and 30 (type on keyboard) form the first group. People performing these activities are usually inclined over one of several objects, i.e., a phone, a book, a notebook or a tablet. The skeletal system of these persons is quite similar. The second group of activities consists of numbers 10 (clapping) and 34 (rub hands). For them, the people's stature is quite similar. The last group consists of activities numbered 31 and 32, i.e., pointing with a finger and taking a selfie. These actions may seem to be much different. However, it should be known that the simplified skeleton model does not have a representation of fingers. Thus, these two behaviors are observed as putting a hand in front of a person. The accuracy of the network with 40 classes was, in this particular case, equal to 89.6%. Let us combine the three subsets of "similar" classes into separate meta-classes ($M1, M2, M3$): $\{11, 12, 29, 30\} \rightarrow M1$, $\{10, 34\} \rightarrow M2$, $\{31, 32\} \rightarrow M3$. Thus, we get a 35-class problem (32 "normal" classes and 3 meta-classes). The mean accuracy of such a classification problem would increase to 93.5%.

Table 8. Comparison of our best solutions with related work for the NTU-RGB+D dataset (40 action classes, CV mode).

Method (Ref. No., Year)	Test Accuracy (%)	Parameters	Sequence
ST-LSTM ([30], 2016)	77.7	>2 M	8
3S-CNN ([33], 2019)	93.7	*unknown*	32
PoseC3D ([38], 2021)	97.1	6.9 M	all
RGBPose-Conv3D ([39], 2022)	99.6	>10 M	all
MPE-act-PSM	83.2	412 k	32
SC-LSTM-act-PS	90.8	3.32 M	32
MPE-act-RAW	75.4	412 k	32
SC-LSTM-act-RAW	79.8	3.32 M	32

Experiment with the UTKinect Dataset Finally, we also ran a cross-dataset experiment using an NTU RGB+D pre-trained SC-LSTM-act for testing on the UTKinect set. A problematic issue when using both collections is that the second set contains only 5 image sequences and 10 action types. For this reason, a modified model was pre-trained on the NTU RGB+D set, where the last layer has been changed from 40 classes of actions to 10 classes. The UTKinect set was divided into a training and test set in a ratio of 9:1. No validation subset was created. The training process has been limited to 50 epochs. The number of network weights to be trained was 397 k. The averaged results of five tries with different training/test splits are as follows: training accuracy 98%, test accuracy 90%.

4.4. Discussion

The proposed solutions to human activity classification were experimentally validated on two video datasets. This use of popular datasets allowed a performance comparison of our approach with other methods described in the literature. The performance of our light-weight solutions (MPE), based on a mixture of pose classifiers, is slightly lower than the best reported results (by up to 10% for comparable 2D skeleton data), but our model is more than 10 times lighter. It still performs similar to or better than relatively old heavy solutions. The performance of our mid-weight solutions (SC-LSTM) is comparable with the current best reported results of similar complexity. Only the top-most solutions, based on Graph CNNs and when exploring dense sequences (all video frames) overpower our results by up to 6%.

5. Summary

Two light-weight and mid-weight models were proposed for human activity classification in sparse image sequences (key frames of video clips). They use as a preliminary step a human skeleton estimation in single frames. Our main focus was to improve the quality of skeleton data and to define relational information for skeletons, which allows us to use simple encoding/classification networks and reach reasonable accuracy. It was experimentally shown, that using our relational features an accuracy improvement of 8-10% has been achieved, compared to the use of RAW skeleton data. Another benefit of our approach is its lightness, which makes it easily applicable on mobile devices and on robotic platforms. A practical advantage is the assumed sparsity of video frames. By adjustment of the key frame number, it makes real-time processing possible even with moderate computational resources. The approach can easily be adopted to process true image sequences, like image galleries.

The limitations of this study are: a focus on the actions of main body parts and the use of a single performance measure:

- As the feature vector is based on the subset of the 15 most reliably detected skeleton joints, human actions performed mainly by feet, hands and fingers, which are not included in this subset, cannot be properly distinguished from each other.
- The evaluation process of the proposed approach could include other popular measures, such as the precision-recall curve and AUC.

Our future work should focus on more extensive training and testing of various network architectures (e.g., on the NTU RGB+120 dataset) and on the extension of feature engineering to deal with partial skeleton information.

AUTHORS

Włodzimierz Kasprzak* – Warsaw University of Technology, Institute of Control and Computation Eng. ul.Nowowiejska 15/19, 00-665 Warsaw, Poland, e-mail: wlodzimierz.kasprzak@pw.edu.pl, Orcid: 0000-0002-4840-8860, www.ia.pw.edu.pl/~wkasprza.

Paweł Piwowarski – Warsaw University of Technology, Institute of Control and Computation Eng. ul.Nowowiejska 15/19, 00-665 Warsaw, Poland, e-mail: pawel@piwowarski.com.pl, Orcid: 0000-0002-4477-2534.

*Corresponding author

ACKNOWLEDGEMENTS

This work was conducted within the project APAKT, supported by "Narodowe Centrum Badań i Rozwoju", Warszawa, grant No. CYBERSECIDENT/455132/III/NCBR/2020.

References

[1] C. Coppola, S. Cosar, D. R. Faria, and N. Bellotto. "Automatic detection of human interactions from RGB-D data for social activity classification," *2017 26th IEEE International Symposium on Robot and Human Interactive Communication "RO-MAN"*, Lisbon, 2017, pp. 871–876; doi: 10.1109/ROMAN.2017.8172405.

[2] A. M. Zanchettin, A. Casalino, L. Piroddi, and P. Rocco. "Prediction of Human Activity Patterns for Human–Robot Collaborative Assembly Tasks," *IEEE Transactions on Industrial Informatics*, vol. 15(2019), no. 7, pp. 3934–3942; doi: 10.1109/TII.2018.2882741.

[3] Z. Zhang, G. Peng, W. Wang, Y. Chen, Y. Jia, and S. Liu. "Prediction-Based Human-Robot Collaboration in Assembly Tasks Using a Learning from Demonstration Model," *Sensors*, 2022, no. 22(11):4279; doi: 10.3390/s22114279.

[4] M. S. Ryoo. "Human activity prediction: Early Recognition of Ongoing Activities from Streaming Videos," *2011 International Conference on Computer Vision*, Barcelona, Spain, 2011, pp. 1036–1043; doi: 10.1109/ICCV.2011.6126349.

[5] K. Viard, M. P. Fanti, G. Faraut, and J.-J. Lesage. "Human Activity Discovery and Recognition using Probabilistic Finite-State Automata. "*IEEE Transactions on Automation Science and Engineering*, vol. 17 (2020), no. 4, pp. 2085–2096; doi: 10.1109/TASE.2020.2989226.

[6] S. Zhang, Z. Wei, J. Nie, L. Huang, S. Wang, and Z. Li. "A review on human activity recognition using vision-based method," *Journal of Healthcare Engineering*, Hindawi, vol. 2017, Article ID 3090343; doi: 10.1155/2017/3090343.

[7] A. Stergiou and R. Poppe. "Analyzing human-human interactions: a survey," *Computer Vision and Image Understanding*, Elsevier, vol. 188 (2019), 102799; doi: 10.1016/j.cviu.2019.102799.

[8] A. Bevilacqua, K. MacDonald, A. Rangarej, V. Widjaya, B. Caulfield, and T. Kechadi. "Human Activity Recognition with Convolutional Neural Networks," *Machine Learning and Knowledge Discovery in Databases (ECML PKDD 2018)*, LNAI vol. 11053, Springer, Cham, Switzerland, 2019, pp. 541–552; doi: 10.1007/978-3-030-10997-4_33.

[9] M. Liu, and J. Yuan. "Recognizing Human Actions as the Evolution of Pose Estimation Maps," *2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2018)*, Salt Lake City, UT, USA, June 18-22, 2018, pp. 1159–1168; doi: 10.1109/CVPR.2018.00127.

[10] E. Cippitelli, E. Gambi, S. Spinsante, and F. Florez-Revuelta. "Evaluation of a skeleton-based method for human activity recognition on a large-scale RGB-D dataset," *2nd IET International Conference on Technologies for Active and Assisted Living (TechAAL 2016)*, London, UK, 2016; doi: 10.1049/IC.2016.0063.

[11] Z. Cao, G. Hidalgo, T. Simon, S.-E. Wei, and Y. Sheikh, "OpenPose: Realtime Multi-Person 2D Pose Estimation Using Part Affinity Fields," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(1):172–186, 2021; doi: 10.1109/TPAMI.2019.2929257.

[12] A. Toshev, and C. Szegedy. "DeepPose: Human Pose Estimation via Deep Neural Networks," *2014 IEEE Conference on Computer Vision and Pattern Recognition*, Columbus, OH, USA, 2014, pp. 1653–1660; doi: 10.1109/CVPR.2014.214.

[13] E. Insafutdinov, L. Pishchulin, B. Andres, M. Andriluka, and B. Schiele. "Deepercut: a deeper, stronger, and faster multi-person pose estimation model," *Computer Vision – ECCV 2016*, LNCS vol. 9907, Springer, Cham, Switzerland, 2016, pp. 34–50; doi: 10.1007/978-3-319-46466-4_3.

[14] [Online]. NTU RGB+D 120 Dataset. Papers With Code. Available online: https://paperswithcode.com/dataset/ntu-rgb-d-120 (accessed on 30 June 2022).

[15] M. Perez, J. Liu, and A.C. Kot, "Interaction Relational Network for Mutual Action Recognition," arXiv:1910.04963 [cs.CV], 2019; https://arxiv.org/abs/1910.04963 (accessed on 15.07.2022).

[16] L.-P. Zhu, B. Wan, C.-Y. Li, G. Tian, Y. Hou, and K. Yuan. "Dyadic relational graph convolutional networks for skeleton-based human interaction recognition," *Pattern Recognition*, Elsevier, vol. 115, 2021, p. 107920; doi: 10.1016/j.patcog.2021.107920.

[17] R.-A. Jacobs, M.-I. Jordan, S.-J. Nowlan, and G.-E. Hinton. "Adaptive mixtures of local experts," *Neural Comput.*, 3(1):79–87, 1991.

[18] S. Puchała, W. Kasprzak, and P. Piwowarski. "Feature engineering techniques for skeleton-based two-person interaction classification in video," *17th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, Singapore, 2022, IEEE Explore, pp. 66–71; doi: 10.1109/ICARCV57592.2022.10004329.

[19] P.-F. Felzenszwalb, R.-B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2010, vol. 32, no. 9, pp. 1627–1645; doi: 10.1109/TPAMI.2009.167.

[20] A. Krizhevsky, I. Sutskever, and G.-E. Hinton, "ImageNet classification with deep convolutional neural networks," *Communications of the ACM*, 2017, vol. 60(6), pp. 84–90; doi: 10.1145/3065386.

[21] K. Simonyan, and A. Zisserman. "Very Deep Convolutional Networks for Large-Scale Image

Recognition," *arXiv*, 2015, arXiv:1409.1556; https://arxiv.org/abs/1409.1556.

[22] K. He, X. Zhang, S. Ren, and J. Sun. "Deep Residual Learning for Image Recognition," *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, 27–30 June 2016, pp. 770–778; doi: 10.1109/CVPR.2016.90.

[23] T.-L. Munea, Y.-Z. Jembre, H.-T. Weldegebriel, L. Chen, C. Huang, and C. Yang. "The Progress of Human Pose Estimation: A Survey and Taxonomy of Models Applied in 2D Human Pose Estimation," *IEEE Access*, 2020, vol. 8, pp. 133330–133348; doi: 10.1109/ACCESS.2020.3010248.

[24] K. Wei, and X. Zhao. "Multiple-Branches Faster RCNN for Human Parts Detection and Pose Estimation," *Computer Vision – ACCV 2016 Workshops*, Lecture Notes in Computer Science, vol. 10118, Springer, Cham, 2017; doi: 10.1007/978-3-319-54526-4.

[25] Z. Su, M. Ye, G. Zhang, L. Dai, and J. Sheng. "Cascade feature aggregation for human pose estimation," *arXiv*, 2019, arXiv:1902.07837; https://arxiv.org/abs/1902.07837.

[26] H. Meng, M. Freeman, N. Pears, and C. Bailey. "Real-time human action recognition on an embedded, reconfigurable video processing architecture," *J. Real-Time Image Proc.*, vol. 3, 2008, no. 3, pp. 163–176; doi: 10.1007/s11554-008-0073-1.

[27] K.-G. Manosha Chathuramali, and R. Rodrigo. "Faster human activity recognition with SVM," *International Conference on Advances in ICT for Emerging Regions (ICTer2012)*, Colombo, Sri Lanka, 12–15 December 2012, IEEE, 2012, pp. 197–203; doi: 10.1109/icter.2012.6421415.

[28] X. Yan, and Y. Luo. "Recognizing human actions using a new descriptor based on spatial–temporal interest points and weighted-output classifier," *Neurocomputing*, Elsevier, vol. 87, 2012, pp. 51–61, 15 June 2012; doi: 10.1016/j.neucom.2012.02.002.

[29] R. Vemulapalli, F. Arrate, and R. Chellappa. "Human Action Recognition by Representing 3D Skeletons as Points in a Lie Group," *2014 IEEE Conference on Computer Vision and Pattern Recognition*, 23–28 June 2014, Columbus, OH, USA, IEEE, pp. 588–595; doi: 10.1109/cvpr.2014.82.

[30] J. Liu, A. Shahroudy, D. Xu, and G. Wang, "Spatio-Temporal LSTM with Trust Gates for 3D Human Action Recognition," *Computer Vision – ECCV 2016*, Lecture Notes in Computer Science, vol. 9907, Springer, Cham, Switzerland, 2016, pp. 816–833; doi: 10.1007/978-3-319-46487-9_50.

[31] A. Shahroudy, J. Liu, T.-T. Ng, and G. Wang. "NTU RGB+D: A Large Scale Dataset for 3D Human Activity Analysis," arXiv:1604.02808[cs.CV], 2016; https://arxiv.org/abs/1604.02808.

[32] C. Li, Q. Zhong, D. Xie, and S. Pu. "Skeleton-based Action Recognition with Convolutional Neural Networks," *2017 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*, 10–14 July 2017, Hong Kong, pp. 597–600; doi: 10.1109/ICMEW.2017.8026285.

[33] D. Liang, G. Fan, G. Lin, W. Chen, X. Pan, and H. Zhu. "Three-Stream Convolutional Neural Network With Multi-Task and Ensemble Learning for 3D Action Recognition," *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 16–17 June 2019, Long Beach, CA, USA, IEEE, pp. 934–940; doi: 10.1109/cvprw.2019.00123.

[34] S. Yan, Y. Xiong, and D. Lin. "Spatial Temporal Graph Convolutional Networks for Skeleton-Based Action Recognition," arXiv:1801.07455 [cs.CV], 2018; https://arxiv.org/abs/1801.07455, (accessed on 15.07.2022).

[35] M. Li, S. Chen, X. Chen, Y. Zhang, Y. Wang, and Q. Tian, "Actional-Structural Graph Convolutional Networks for Skeleton-Based Action Recognition," *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, USA, 15–20 June 2019, pp. 3590–3598; doi: 10.1109/CVPR.2019.00371.

[36] L. Shi, Y. Zhang, J. Cheng, and H.-Q. Lu. "Two-Stream Adaptive Graph Convolutional Networks for Skeleton-Based Action Recognition," arXiv:1805.07694v3 [cs.CV], 10 July 2019; doi: 10.48550/ARXIV.1805.07694.

[37] L. Shi, Y. Zhang, J. Cheng, and H.-Q. Lu. "Skeleton-based action recognition with multi-stream adaptive graph convolutional networks," *IEEE Transactions on Image Processing*, vol. 29, October 2020, pp. 9532–9545; doi: 10.1109/TIP.2020.3028207.

[38] H. Duan, Y. Zhao, K. Chen, D. Shao, D. Lin, and B. Dai. "Revisiting Skeleton-based Action Recognition," *arXiv*, 2021, arXiv:2104.13586; https://arxiv.org/abs/2104.13586.

[39] H. Duan, Y. Zhao, K. Chen, D. Lin, and B. Dai. "Revisiting Skeleton-based Action Recognition," arXiv:2104.13586v2 [cs.CV], 2 Apr 2022; https://arxiv.org/abs/2104.13586v2.

[40] J. Liu, G. Wang, P. Hu, L.-Y. Duan, and A. C. Kot. "Global Context-Aware Attention LSTM Networks for 3D Action Recognition," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, 21-26 July 2017, pp. 3671–3680; doi: 10.1109/CVPR.2017.391.

[41] J. Liu, G. Wang, L.-Y. Duan, K. Abdiyeva, and A. C. Kot. "Skeleton-Based Human Action Recognition with Global Context-Aware Attention LSTM Networks," *IEEE Transactions*

on Image Processing (TIP), 27(4):1586–1599, 2018; doi: 10.1109/TIP.2017.2785279.

[42] J. Liu, A. Shahroudy, G. Wang, L.-Y. Duan, and A. C. Kot. “Skeleton-Based Online Action Prediction Using Scale Selection Network,” *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 42(6):1453–1467, 2019; doi: 10.1109/TPAMI.2019.2898954.

[43] T. Yu, and H. Zhu. “Hyper-Parameter Optimization: A Review of Algorithms and Applications,” arXiv:2003.05689 [cs, stat], 2020; https://arxiv.org/abs/2003.05689.

[44] [Online]. “openpose”, CMU-Perceptual-Computing-Lab, 2021; https://github.com/CMU-Perceptual-Computing-Lab/openpose/.

[45] [Online]. “Keras: the Python deep learning API,” https://keras.io/.

[46] [Online]. “UTKinect-3D Database,” Available online: http://cvrc.ece.utexas.edu/KinectDatasets/HOJ3D.html (accessed on 30 June 2022).

[47] Kiwon Yun. “Two-person Interaction Detection Using Body-Pose Features and Multiple Instance Learning,” https://www3.cs.stonybrook.edu/~kyun/research/kinect_interaction/index.html.