DOI: 10.15439/2022F195
ISSN 2300-5963 ACSIS, Vol. 30

A lightweight approach to two-person interaction classification in sparse image sequences

Włodzimierz Kasprzak, Paweł Piwowarski
Warsaw University of Technology
Institute of Control and Computation Eng.
ul. Nowowiejska 15/19
00-665 Warszawa, Poland
Email: {wlodzimierz.kasprzak, pawel.piwowarski.dokt}@pw.edu.pl

Van-Khanh Do
no affiliation
Email: khanhdovanit@gmail.com

***Abstract*—A lightweight neural network-based approach to two-person interaction classification in image sequences, based on human skeletons detected in sparse video frames, is proposed. The idea is to use an ensemble of pose classifiers (”experts”), where every expert is trained on different time-indexed snapshots of an interaction. Thus, the expertise of "weak" classifiers is distributed over the time duration of an interaction. The overall classification result is a weighted combination of all the pose experts. Important element of proposed solution is the refinement of skeleton data, based on a merging-of-joints procedure. This allows the generation of reliable features being passed to the artificial neural network. This is the key to our lightweight solution, as ANN resources, needed for feature space transformation, can be significantly limited. Our network model was trained and tested on the interaction subset of the well-known NTU RGB+D dataset, although only 2D skeleton information is used, typical in video analysis. The test results show comparable performance of our method with some of the best so far reported STM- and CNN-based classifiers for this dataset, when they process sparse frame sequences, like we did. The recently proposed multi-stream Graph CNNs have shown superior results but only when processing dense frame sequences. Considering the dominating processing time and resources needed for skeleton estimation in every frame of the sequence, the key to real-time interaction recognition is to limit the number of processed frames.**

I. Introduction

The aim of our work is the analysis of human interactions in specific time-related image sequences. The data can originate from decomposition of video clips onto frames or directly from snapshots of videos posted as image galleries in the Internet. Their common property is the sparsity of time-relevant information (Figure 1).

The approaches to vision-based human activity recognition can be divided into two main categories: activity recognition directly in video data [1] or skeleton-based methods [2], where the 2D or 3D human skeletons are detected first, even by specialized devices, like the Microsoft Kinect.

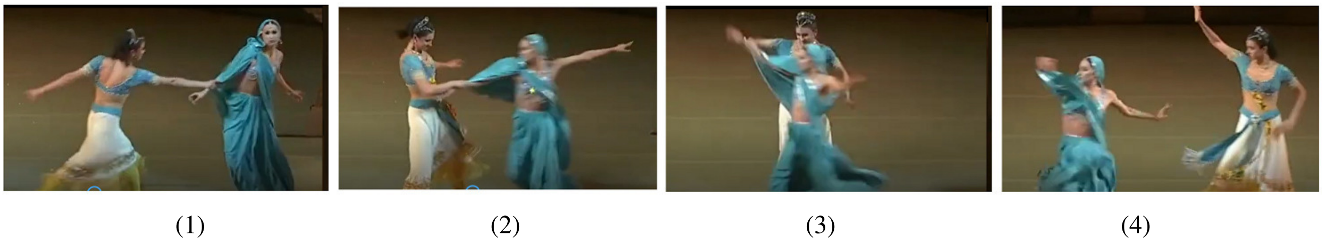
In early solutions, hand-designed features like edges, contours, Scale-Invariant Feature Transform (SIFT), Histogram of Oriented Gradients (HOG) have usually been used for detection and localization of human body parts or key points in the image [3], [4].

More recently, Neural Network-based solutions were successfully proposed for solving human pose- and human activity recognition problems, e.g., solutions are based on Deep Neural Networks (DNN) [5], especially on Long-Short Term Memory (LSTM) models and Convolutional Neural Networks (CNN) [6], and more recently on Graph CNNs [7]. CNNs have the capability automatically to learn rich semantic and discriminative features from images and multi-dimensional signals. Furthermore, CNNs can learn both spatial and temporal information from signals and model scale-invariant features as well. Graph CNNs allow efficient implementations of convolution layers when structured data (i.e., graphs) are processed. Some popular solutions to human skeleton estimation (i.e., the detection and localization) in images, based on DNN and CNN models, can be mentioned: OpenPose [8], DeepPose [9] and DeeperCut [10].

Hence, nowadays human action- and interaction recognition in video is most often based on skeleton data extracted from video frames. The state-of-the-art solutions to human action encoding and classification, which process human skeleton data, typically use "heavy" deep neural networks, like 3D CNNs and LSTMs or slightly lightweight Graph CNNs [11], [12].

In this work, we focus on two-person interaction recognition in sparse frame sequences, assuming the existence of skeleton data for key video frames. We took the straightforward idea of extending two-person pose classification of still images to two-person interaction classification in image sequences, by applying an ensemble of pose classifiers [13]. Typically for a classifier ensemble, individual classifiers are ”experts” in different parts of the input data domain and the extra weighting network differentiates between subdomains. In our approach, the pose-classifiers are experts at different time stages, while their input space itself (i.e., the spatial image information) is not affecting the fusion weights. By performing a simple time decomposition, we are going to distinguish four subsequent time periods of an interaction process, e.g. start, before midterm, after midterm and final. The final fusion will take the form of a weighted sum of class likelihoods of all the pose classifiers.

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



(1) (2) (3) (4)

Fig. 1. Example of a sparse sequence of frames from a two-person interaction video

There are 4 remaining sections of this work. Section II refers some recent approaches in human pose, -action and -interaction recognition. Our solution is presented in section III. In section IV, experiments are described, to verify the approach. The classifiers are trained and tested on two datasets: an own human pose image dataset, called ”humiact5”, and the well-known video dataset for action and interaction, NTU RGB+D [14]. Finally, in section V, we summarize our work and contribution to the subject.

II. Related work

The recognition of human activities in video is a hot research topic in the last 15 years. 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 lightness sensitivity.

The vast majority of research is based on the use of artificial neural networks. However, more classical approaches have also been tried, such as the SVM (e.g. [15], [16]). Yan et al. [17] 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. [18] human actions are modeled as curves in a Lie group of Euclidean distances. The classification process is using 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. [19] and Shahroudy et al. [14]. 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 [20] 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. [21] 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 an own stream but after every stage the results are exchanged between streams.

Graph convolutional networks are currently considered as 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” [22] and ”Actional-Structural Graph Convolutional Networks” [23] are examples of such an solution.

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” (AAGCN), proposed by Shi et al. [24], [25].

One of the best performances on the NTU RGB+D interaction dataset is reported in the work of Perez et al. [26]. Its main contribution is a powerful two-stream network with three-stages, called ”Interaction Relational Network” (IRN).

The network input 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 are reported by Zhu et al. [27], 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.

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, 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.

III. The approach

A. Structure

The input data for our interaction classifier is a sequence of sparse video frames. Assuming, a video clip is given the start and end of an interaction should be detected first. Then, the video clip is split into some number M of consecutive time intervals (e.g. $M = 16$). From each interval one frame is selected for classification. Assume, that $M = N \cdot m$, where N is a period of time, while m the number of frames in one period. We may distinguish $N = 4$ periods: start, 1-st intermediate, 2-nd intermediate and final. To the classification of frames from a single period, a separate pose classifier (the "expert") is dedicated. As shown in Figure 2, the proposed solution consists of several processing stages:

1) *Skeleton estimation:* the OpenPose net [28] is applied to detect human skeletons with their 2D joints in an RGB image (a video frame);
2) *Feature engineering:* a *keypoint enhancement algorithm* is proposed in order to get more reliable two sets of skeleton joints from the OpenPose results; next, *feature vectors* are extracted from the refined joints.
3) *Pose classifier training:* several lightweight, densely-connected MLP networks are trained - every one is a "weak" classifier.
4) *Model evaluation:* alternative network models are evaluated, to find the optimal model configuration and training parameter. A *Keras*-tuner [29] - the *RandomSearch* algorithm [30] is applied to find optimal hyper-parameter settings.
5) *Ensemble classifier:* a dense gain network is also trained to learn the weights for results of individual pose classifiers. Two versions of the final classifier are implemented - one with fixed weights and one with learned weights.
6) *Model testing:* after accumulating the pose class likelihoods over the frame sequence the final most likely interaction class is selected as the winner. Two datasets - an own humiact5 and the RGB subset of the NTU RGB+D dataset, are used to evaluate the created models.

B. Skeleton estimation

In the paper [8], a multi-person 2D pose estimation architecture was proposed based on part affinity fields (PAFs). The work introduced an explicit nonparametric representation of the keypoint association which encodes both position and orientation of the human limbs. The designed architecture can learn both human keypoint detection and association using heatmaps of human key-points and part affinity fields respectively. It iteratively predicts part affinity fields and part detection confidence maps. The part affinity fields encode part-to-part association including part locations and orientations. In the iterative architecture, both PAFs and confidence maps will be iteratively refined over successive stages with intermediate supervision at each stage. Subsequently, a greedy parsing algorithm is employed to effectively parse human poses. The work ended up releasing the OpenPose library, the first real-time system for multi-person 2D pose estimation [28]. In our research, we use the core block of OpenPose, the "body_25 model", to extract 25 human key-points in images. The result is an 25-elementary array, providing 2D image coordinates and confidence score for every keypoint.

C. Feature engineering

From the (eventually more than two) sets of skeleton joints, detected in the image by OpenPose, the two main actors are selected based on size measure. A total variability of skeleton keypoint locations is calculated for every skeleton and the two with the highest variability are chosen for feature engineering.

1) Skeleton enhancement: There are cases where OpenPose wrongly splits one human region into different regions due to

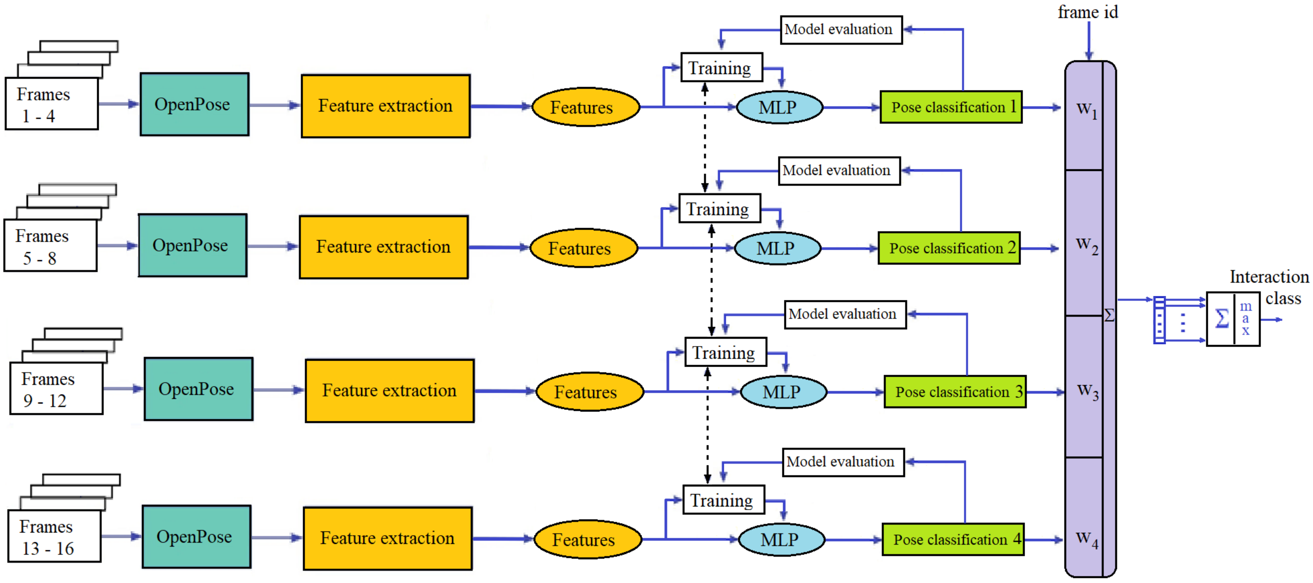


Fig. 2. General structure of our approach

occlusion, low resolution, or complex visual context. Therefore, we developed a keypoint (i.e., skeleton joints) merging and replacement algorithm. In the first step, we try to merge sets of joints, where applicable, to produce finer skeleton joints (see Figure 3).

Two calculations are made for each pair of sets including the number of intersection points of the two sets and the distance between them. The intersection indicator value is scaled by the number of points of the smaller set. The distance calculation takes their two mean points and two standard deviation values into account. These calculated values then will be compared with corresponding thresholds to decide whether the two sets are going to be merged or not. In case merging conditions are met, the intersection points of the two sets will be treated in the following way: the data points with higher probability will be kept and the lower ones will be ignored.

For the sake of clarity, Figure 4 illustrates the merging procedure of two specific sets, A and B, based on the assumption that they come from the same person in the image. The bigger set A is missing key-points for the left leg, while the smaller set B includes these key-points. The mean points (center of gravity) of A and B are m_A and m_B, respectively, the standard deviations of joints locations for A and B are $[std_{A,x},\ std_{A,y}]$ and $[std_{B,x},\ std_{B,y}]$, respectively.

The conditions for a merging action are as follows:

$$\frac{|A \cap B|}{|B|} \leq \theta_1 \tag{1}$$

$$\frac{|m_{A,x}\ -\ m_{B,x}|}{std_{A,x}\ +\ std_{B,x}} + \frac{|m_{A,y}\ -\ m_{B,y}|}{std_{A,y}\ +\ std_{B,y}} \leq \theta_2 \tag{2}$$

where θ_1, θ_2 are intersection threshold and distance threshold, respectively.

After a merging action has been performed, the remaining joints in the smaller set (call it S) can eventually replace low-confident, corresponding joints in a subset B_s of the big set B. To decide about this, the following values are considered: the normalized Euclidean distance between the smaller set joints and the corresponding candidate joints of the subset B_s, the average confidence of all candidate joints in the small set S and the average confidence of the corresponding joints in the bigger set (Figure 5).

Let N-elementary sets S and B_s of corresponding joints are given, considered for possible replacement. Standard deviation coefficients of the smaller set joints locations are $[std_{S,x}, std_{S,y}]$. Let the confidence value of a joint j be denoted as $P(j)$. The conditions for a joints replacement are as follows:

$$\frac{1}{std_{S,x} + std_{S,y}} \sum_{i=1}^{N} \sqrt{(x_{S_i} - x_{B_{s_i}})^2 + (y_{S_i} - y_{B_{s_i}})^2} \leq \theta_3 \tag{3}$$

$$\frac{1}{N} \sum_{i=1}^{N} P(S_i) \geq \theta_4 \tag{4}$$

$$\frac{1}{N} \sum_{i=1}^{N} P(B_{s_i}) \leq \theta_5 \tag{5}$$

where θ_3 is the normalized Euclidean distance threshold, θ_4 - the confidence threshold for S and θ_5 - the confidence threshold for B_s.

The skeletons, which remain after the merging and replacement steps, will be ordered by their bounding box size in descending order. With (w, h) representing width and height of a bounding box, the $score = w{\cdot}h$. The two sets with highest score will be kept and used further in the feature extraction step.

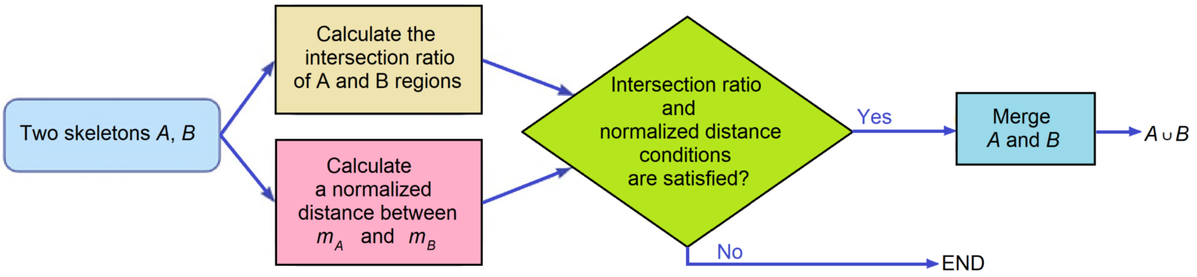

Two skeletons A, B
Calculate the intersection ratio of A and B regions
Calculate a normalized distance between m_A and m_B
Intersection ratio and normalized distance conditions are satisfied?
Yes
No
Merge A and B
A ∪ B
END

Fig. 3. The skeleton merging step.

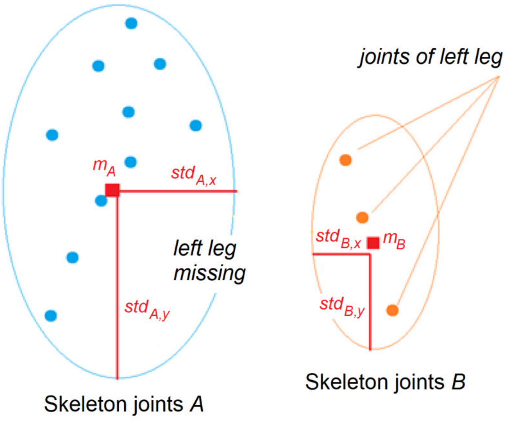


Fig. 4. Illustration of a skeleton merging situation.

2) Feature extraction: Feature extraction means the calculation of normalized distances between pairs of joints from two skeletons, tracked in the frame sequence of a video clip. First, the distance between two middle-of-spine points of two human skeletons is calculated and normalized by the length of the spine I_1 of the first person, giving the distance feature (Figure 6). Then, every set of joints is independently normalized by: translating the local coordinate system to the middle-of-spine point O_1 or O_2, rotating the points so that the spine segment (connecting joint 1 with joint 8) is parallel to the Y axis of local system, and finally, scaling the point coordinates by the spine length I_i.

Denote by $\mathbf{H}_1, \mathbf{H}_2$ the skeletons of the first and the second human; O_1, O_2 - the centers of spine segments of the first and second human, respectively; l_1 - the length of the spine segment of the first human; α_1, α_2 - the rotation angles to make corresponding spine segments parallel to the Y axes of local Cartesian coordinate systems. The *distance* feature is calculated as the distance between local system origins, O_1, O_2, normalized by the length l_1:

$$d = \frac{distance}{l_1} \tag{6}$$

The normalization of joints coordinates (translation to local system, rotation, scaling) is performed independently for every set $\mathbf{H}_1, \mathbf{H}_2$. Let $\mathbf{p}_i = (p_{i,x}, p_{i,y})$ denotes the image coordinates of a joint from skeleton $\mathbf{H}_i, (i = 1, 2)$. The normalization of this joint is given as follows:

$$\mathbf{p}'_i = (p'_{i,x}, p'_{i,y}) = (p_{i,x} - O_{i,x}; p_{i,yi} - O_{i,y}),\ i = 1, 2 \tag{7}$$

$$\begin{pmatrix} p''_{i,x} \\ p''_{i,y} \end{pmatrix} = \begin{bmatrix} \cos(\alpha_i) & -\sin(\alpha_i) \\ \sin(\alpha_i) & \cos(\alpha_i) \end{bmatrix} \begin{pmatrix} p'_{i,x} \\ p'_{i,y} \end{pmatrix},\ i = 1, 2 \tag{8}$$

$$(p'''_{i,x}, p'''_{i,y}) = \left(\frac{p''_{i,x}}{w_i}, \frac{p''_{i,y}}{h_i} \right),\ i = 1, 2 \tag{9}$$

3) Feature vector: Both the OpenPose (applied for our RGB dataset) and the built-in skeleton detector from Kinect v2 (generating the skeleton data in the NTU RGB+D dataset) deliver person skeletons of 25 joints. By analysing a small skeleton data subset, we found that the data for joints numbered from 15 to 24, corresponding to ”small” parts, like fingers, are very often missing. Thus, we use only joints numbered from 0 to 14. The feature vector obtained from skeleton data of a single frame has 61 dimensions as there are $15\ joints \times 2\ coordinates \times 2\ sets$ and one distance feature. Assuming that we have selected m frames for analysis, we get a map of $m \times 61$ features.

D. Pose classifier training and evaluation

The feature data is fed to several MLP-based pose classifiers. We use fully-connected MLP architecture with variants of several hyper-parameters: 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 ANN is implemented using Keras [29].

Automated hyper-parameter tuning [31] is a crucial step during ANN model training to increase the model's performance. We perform a hyper-parameter search during training using the *Random Search* algorithm, offered in Keras [30]. For both datasets the hyper-parameter search space is defined as:

$$S_{search} = [a_{fun}, l_{rate}, n_{layer}, n_{neur}], \tag{10}$$

where the entries are: activation function, $a_{fun} \in \{relu, sigmoid\}$, learning rate, $l_{rate} \in [10^{-5}, 10^{-2}]$, number of hidden layers, $n_{layer} \in \{1, 2, 3\}$, number of neurons in hidden layer, $n_{neur} \in \{100, 200, \ldots, 1000\}$.

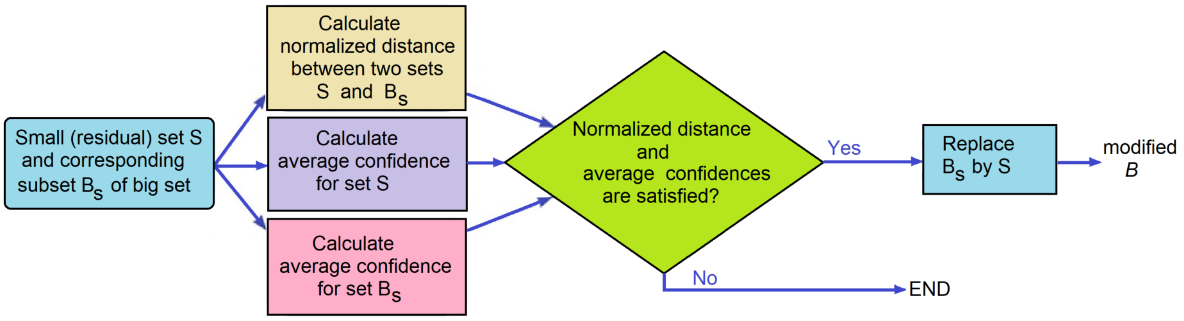

Small (residual) set S and corresponding subset B$_S$ of big set
Calculate normalized distance between two sets S and B$_S$
Calculate average confidence for set S
Calculate average confidence for set B$_S$
Normalized distance and average confidences are satisfied?
Yes
No
Replace B$_S$ by S
modified B
END

Fig. 5. The joints replacement step

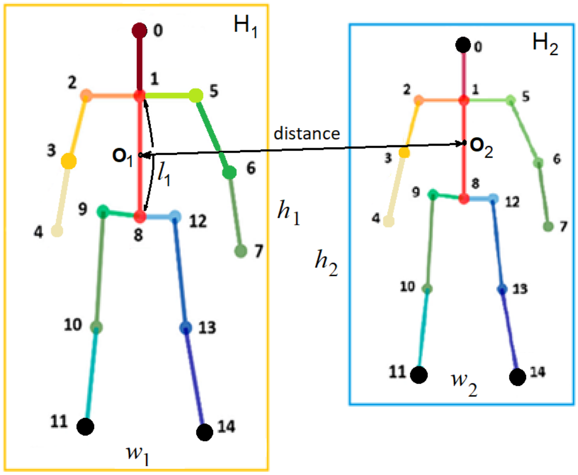


Fig. 6. The normalization elements for two sets of joints

E. Ensemble classifier

As mentioned earlier, every pose classifier is an "expert" to recognize snapshots taken during different time period of an interaction. In practice, the training of such an assembly is performed at the same time, but 3 out of 4 "expert" networks are always in a dropout mode. The actually updated network depends on the time period the current input frame belongs to.

In the testing process, the interaction class is known after the entire frame sequence - from a single video clip - has been classified and the results of individual pose classifiers were accumulated. The likelihood of every interaction class comes from an aggregation of pose class likelihoods, as a weighted sum of pose likelihoods, for frames indexed from t=0 to t=T.

1) Fixed gains: In a hand-crafted form we define the aggregation of likelihoods, obtained by particular pose classifiers $(i = 1, 2, 3, 4)$ for frames $(t = 1, 2, ..., N)$, the $\mathbf{Pr}_{pose\_i}(t) - s$, as follows:

$$\begin{aligned}\mathbf{S} = \textstyle\sum_{t=0}^{T} [&\mathbf{Pr}_{pose\_1}(t) \cdot max(0, (T/2 - t)/T) + \\ &+\mathbf{Pr}_{pose\_2}(t) \cdot min(0.5, t/T) + \\ &+\mathbf{Pr}_{pose\_3}(t) \cdot min(0.5, (T - t)/T) + \\ &+\mathbf{Pr}_{pose\_4}(t) \cdot min(0, (t - T/2)/T)]\end{aligned} \tag{11}$$

2) The gain network: In the trained case, the gain network provides gain coefficients $w_i(t)$ for the four pose classifiers depending on the frame index (t):

$$\begin{aligned}\mathbf{S} = \textstyle\sum_{t=0}^{T} [&\mathbf{Pr}_{pose\_1}(t) \cdot w_1(t) + \mathbf{Pr}_{pose\_2}(t) \cdot w_2(t) + \\ &+\mathbf{Pr}_{pose\_3}(t) \cdot w_3(t) + \mathbf{Pr}_{pose\_4}(t) \cdot w_4(t)]\end{aligned} \tag{12}$$

IV. Results

A. Datasets

In order to evaluate and test the trained classifiers, two datasets were used. The search after best hyper-parameters of a single pose classifier will be performed by training and validating them on our **humiact5** dataset. Its consists of images of 5 two-person poses - snapshots of interactions: boxing, facing, hand holding, hand shaking and hugging/kissing. There are 1695 images in total, in which 1154 images are in the training set and remaining 541 images are in the evaluation set (Figure 7). In this series of experiments, the OpenPose library has been applied for skeleton detection in RGB images.

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. 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, A60: walking apart. In our experiments, already the skeleton data of the NTU RGB+D dataset is considered. There are 10420 video clips in total, in which ca. 70% are in the training set and remaining 30% are in the test set. No distinct validation subset is distinguished.

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. We apply the cross-subject (CS) evaluation mode, i.e., videos of 20 persons are used for training and videos of remaining 20 persons - for testing. The training set contains video clips of users identified as: 1, 2, 4, 5, 8, 9, 13, 14, 15, 16, 17, 18, 19, 25, 27, 28, 31,

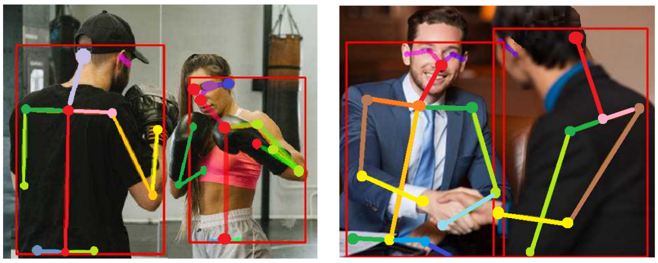


Fig. 7. Samples from our *humiact5* dataset: RGB images with skeleton data

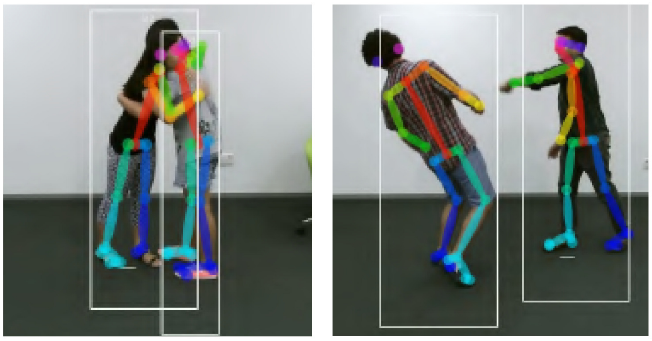


Fig. 8. Samples from the NTU RGB+D interaction dataset: RGB video frames with skeleton data [14]

34, 35 and 38. The number of samples in the training set is 7649, while in the test set - 2771.

Each skeleton instance consists of 25 joints of 3D skeletons that apparently represent a single person (Figure 8). As our research objective is to analyse video data and to focus on only reliably detected joints, we use only the 2D information of only first 15 joints.

From a video sample a set of frames is chosen as follows: the video clip is uniformly split into $N = 4$ time intervals (”periods”), from every interval some number of frames m is selected (we tested $m = 2, 4, 8$). The number of frames in the training set grows from 61192 to 244768 and the number of frames in the test set grows from 22168 to 88672, accordingly to the value of m from 2 to 8.

B. Pose classifier optimization

The hyper-parameter optimization of a pose classifier is performed on the small **humiact5** dataset. In order to run the *RandomSearch* function of Keras, a *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. Using the *RandomSearch* function, we identified three ANN configurations, each one being optimal for given number of hidden layers (1, 2 or 3).

The performances of the three selected models after 100 epochs of training are shown in Table I. The best 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. The learning rate is $5.89 \cdot 10^{-5}$.

TABLE I
THE MEAN ACCURACY ON THE *humiact5* DATASET OF THREE OPTIMAL ANN CONFIGURATIONS WITH 1, 2 AND 3 HIDDEN LAYERS.

Training/test	1 hidden	2 hidden	3 hidden
ANN - mean training accuracy	95%	96%	99%
ANN - mean test accuracy	82%	84%	82%

TABLE II
THE MEAN ACCURACY OF POSE CLASSIFIERS VERIFIED ON THE NTU RGB+D INTERACTION DATASET IN THE CS (CROSS SUBJECT) MODE

Expert	2 f/p	4 f/p	8 f/p
Pose - mean training accuracy	79.2%	82.4%	88.2%
Pose - mean test accuracy	61.2%	70.8%	76.1%

C. Verification on the NTU RGB+D dataset

We train and test our models in the CS (*cross-subject*) verification mode proposed for the NTU RGB+D dataset, i.e. when actors in the training set are different than in the test set, but data from all the camera views are included in both sets. The frame sampling process for both training and testing will be done three times with different number of frames per time period (i.e., extracted from a single video sample): 2, 4, 8. The training set is split into learning and test subsets - two third for learning and one third for validation/testing. There are run 100 epochs of training and the best validation result will be chosen.

1) Pose classifiers: In the follwing, we apply the second version of the ANN pose classifiers, with two hidden layers, as reported earlier in Table I. We train four pose classifiers three times - every one is effectively trained on different frames according to its dedicated time period of action (i.e. $t \in [0, T/4]$, $[T/4, 2T/4]$, $[2T/4, 3T/4]$, $[3T/4, T]$) of training samples with different frame sampling rates (i.e. $n = 2, 4, 8$ frames/period). The mean accuracy of these four pose experts, depending on the number of frames per period is shown on (Table II).

An immediate observation is, that all learning and test accuracies increase, when the training data size is increased. Specifically, with 8 frames per time-period (f/p), these accuracies reach to 88% and 76%, respectively. The average per class accuracies (i.e. four class poses representing the same interaction class) of the ANN experts, obtained with a 4 f/p

TABLE III
THE PER-CLASS TEST ACCURACY OF ANN POSE EXPERTS TRAINED ON THE NTU RGB+D INTERACTION DATASET, VERIFIED IN THE CS (CROSS SUBJECT) MODE, WHEN SAMPLED WITH 4 F/P

Class	A050	A051	A052	A053	A054	A055
Test accuracy	58%	52%	66%	69%	72%	83%
Class	**A056**	**A057**	**A058**	**A059**	**A060**	
Test accuracy	70%	64%	80%	81%	78%	

TABLE IV
THE ACCURACIES OF ANN POSE CLASSIFIER AND TWO VERSIONS OF THE ENSEMBLE CLASSIFIER (E-ANN-1, E-ANN-2), VERIFIED ON THE NTU RGB+D INTERACTION DATASET IN THE CS (CROSS SUBJECT) MODE

Classifier	Training accuracy	Test accuracy
Mean of pose classifiers	88.2%	76.1%
E-ANN-1, eq. (11)	92.4%	81.3%
E-ANN-2, eq. (12)	94.5%	83.3%

TABLE V
THE PER-CLASS TEST ACCURACY OF ANN ENSEMBLE CLASSIFIER, TRAINED ON THE NTU RGB+D INTERACTION DATASET, VERIFIED IN THE CS (CROSS SUBJECT) MODE, WHEN SAMPLED WITH 8 F/P

Class	A050	A051	A052	A053	A054	A055
Test accuracy	67%	67%	77%	87%	86%	91%
Class	**A056**	**A057**	**A058**	**A059**	**A060**	
Test accuracy	81%	76%	92%	93%	90%	

frame sampling on the test set, is shown on Table III. There are 3 classes (A55, A58, A59) that perform at least at 80%, other 6 classes - from 60% to 80% and two - below 60%. Compared with random choice - there are 11 classes and the random prediction (a guess) would be $1/11 = 9.09\%$. The largest accuracy is observed for the ”A055 - hugging” class. The distance between two persons is here significantly smaller than of the rest and the poses are relatively stable in every time period.

2) *Ensemble of pose classifiers:* There are two variants of the final ensemble classifiers: E-ANN-1, when the final score of every interaction class is obtained by fixed weights, according to equation (11), or E-ANN-2, where the trainable gain network is used, according to equation (12). The class with highest score is selected as the winner of the interaction classifier. A notable improvement of interaction classification is observed, when accumulating over time sequence the weighted pose likelihoods. The mean accuracy of the best version of pose experts (i.e., for frame sampling of 8 f/p) was 88.2% (training) and 76.1% (testing), while the ensemble classifier has reached 92.4 % and 81.3% (version 1), or 94.5% and 83.3% (version 2), respectively (Table IV).

The per-class test accuracy of our ensemble classifier E-ANN-2 (with 8 f/p frame sampling) is shown in Table V. There are four classes (A55, A58, A59, A60) with an accuracy of 90% and higher, while the lowest performance (67%) is achieved for classes A50 (punch) and A51 (kicking), The confusion matrix for this testing case is shown in Figure 9. As the numbers of class samples in the test set are slightly unbalanced, we normalized the results, assuming 276 test instances per class, to make them easier comparable. ”Punch” (A50) is most often misclassified with classes A51-A58, which all use hands to express an action, but most often is confused with ”Point finger”. ”Kicking” (A51) is frequently confused with all other classes, slighly less with ”pat on back”. The main errors appear between actions ”giving an object” (A56) and ”shaking hands” (A58) - 18 and 9 cases, and between ”pat on back” (A53) and ”touch pocket” (A57) - 9 and 18 cases.

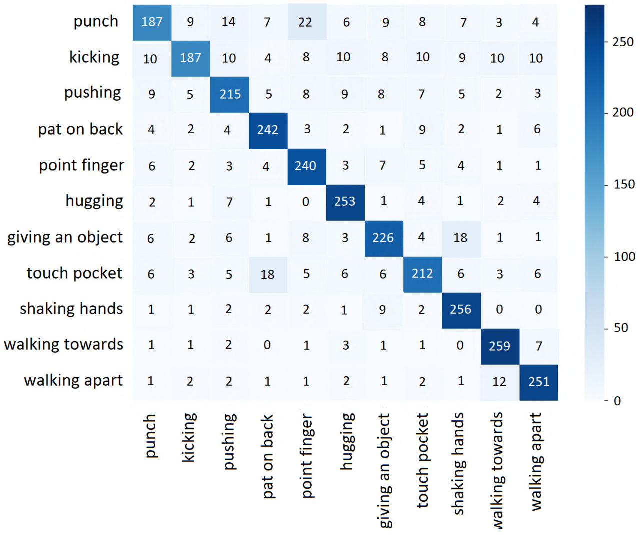


Fig. 9. The confusion matrix for the ensemble classifier E-ANN-2, verified on the NTU RGB+D interaction dataset in the CS (cross subject) mode

TABLE VI
INTERACTION CLASSIFICATION ACCURACY OF LEADING WORKS EVALUATED ON THE NTU RGB+D INTERACTION SET IN THE CS (CROSS SUBJECT) MODE. NOTE: † - RESULT ACCORDING TO [26], ‡ - RESULT ACCORDING TO [27]

Work - reference	Accuracy	Parameters	Frames
FSNET [32]	74.0% (†)	$\sim 200K$	32
ST-LSTM [19]	83.0% (†)	$\sim 2.1M$	32
ST-GCN [22]	83.3% (†)	$3.08M$	32
Our E-ANN-2	83.3%	$400K$	32
GCA-LSTM [33]	85.9% (†)	*unknown*	32
2S GCA-LSTM [34]	87.2% (†)	*unknown*	32
AS-GCN [23]	89.3% (†)	$\sim 9.5M$	32
IRN$_{inter+intra}$ [26]	85.4%	$\sim 9.0M$	32
LSTM-IRN [26]	90.5%	$\sim 9.08M$	$\max(all, 128)$
2S-AGCN [24]	93.4% (‡)	$3.0M$	$\max(all, 300)$
AAGCN [25]	91.5% (‡)	$\sim 6.0M$	$\max(all, 300)$
DR-GCN [27]	93.6%	$3.18M$	$\max(all, 300)$
2S DR-AGCN [27]	94.6%	$3.57M$	$\max(all, 300)$

D. Comparison

Many approaches to two-person interaction classification have been tested on the NTU RGB+D interaction dataset. We list some of the leading works in the Table VI. Our solution needs a low number of weights to be trained and it processes a sparse frame sequence. It shows a good tradeoff between competitive accuracy and low complexity when compared with other recently reported results.

Let us notice how we counted the number of parameters of the E-ANN-2 network. Remember that the pose classifiers have a common part - the feature transforming MLP with 2-hidden layers - and there are separate fully-connected output layers for every pose classifier. We can create two versions of the E-ANN network - one network with multiple feature-transforming MLPs that processes in parallel the four frame subsets, and another one that processes all frames in sequence.

As the individual results are finally aggregated over all frames, both configurations deliver the same final result. In the first configuration, there are 1 597 677 weights needed, while in the sequential version - 399 444 weights only:

1) The feature transforming ANN: $61 \cdot 700 + 700 + 700 \cdot 500 + 500 = 393\ 900$ The FC classification layer: $500 \cdot 11 + 11 = 5\ 511$ The gain network: $(11 + 11) + 11 = 33$
2) Four parallel pose classifiers: $4{\cdot}393\ 900{+}4{\cdot}5\ 511{+}33 = 1\ 597\ 677$
3) Four sequential pose classifiers: $393\ 900{+}4{\cdot}5\ 511{+}33 = 399\ 444$

Taking into account, that the dominating processing time for a single frame is spent by the skeleton detector (on our equipment, it takes ca. 67 ms, compared to 1 ms for the pose classifier), the sequential version is preferred. Even when the skeleton detection itself will be performed in parallel, for every phase subset of frames one pose classifier will be allocated, the sequential version will take only $(N-1)$ ms more time than when using N pose classifiers in parallel.

Typically, the performance of an interaction classifier is significantly improved when dense frame sequences are processed instead of sparse ones. But the overall processing time grows proportionally to the frame number, as the computation is dominated by the skeleton estimation step. Thus, processing a dense sequence of 100 frames (typical for the best performing solutions with accuracy $> 90\%$) takes roughly three times longer than the time needed for a sparse sequence of 32 frames (where a typical accuracy is $< 90\%$). The recently proposed multi-stream Graph CNNs have shown superior results but only when processing dense frame sequences. Considering the dominating processing time and resources needed for skeleton estimation in every frame of the sequence, the key to real-time interaction recognition is to limit the number of processed frames.

V. Conclusion

A light-weight approach to two-person interaction classification was proposed, that can be applied both in video- and single image-analysis. This is a skeleton-based approach, what means, that an external module for human detection and estimation in images is needed. We adopted the state-of-the art OpenPose library for this purpose. This is a powerful deep network solution for human skeleton estimation in images. Our main contribution are algorithms for skeleton data correction and normalization and the design of an ANN classifier that has the form of an ensemble of several ANN-based pose experts. Aggregating four or more "weak" pose classifiers leads to an efficient and robust solution to human interaction classification. We also found that a comparison of classification approaches should not only consider the accuracy measure but also the amount of information received (i.e., whether a sparse or dense frame sequence is analyzed). Our future research should focus on the extraction of motion information for the skeleton joints and testing the model network on longer frame sequences.

References

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

[2] 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," in *2nd IET International Conference on Technologies for Active and Assisted Living (TechAAL 2016)*, London, UK, 24-25 October 2016, pp. 1-6, doi: 10.1049/ic.2016.0063.

[3] 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, 31 pages, 2017, doi: 10.1155/2017/3090343, https://www.hindawi.com/journals/jhe/2017/3090343/

[4] A. Wilkowski, W. Kasprzak and M. Stefanczyk, "Object detection in the police surveillance scenario," in *Proceedings of the 2019 Federated Conference on Computer Science and Information Systems,* ACSIS, vol. 18, 2019, pp. 363-372, doi: 10.15439/2019F291 .

[5] A. Stergiou and R. Poppe, "Analyzing human-human interactions: A survey," *Computer Vision and Image Understanding*, Elsevier, vol. 188, 2019, p. 102799, doi: 10.1016/j.cviu.2019.102799, https://www.sciencedirect.com/science/article/pii/S1077314219301158

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

[7] N. A. Mac and N. H. Son, "Rotation Invariance in Graph Convolutional Networks," in *Proceedings of the 16th Conference on Computer Science and Intelligence Systems*, ACSIS, vol. 25, 2021, pp. 81–90, doi: 10.15439/2021F140 .

[8] Z. Cao, G. Hidalgo, T. Simon, S.-E. Wei, and Y. Sheikh, "OpenPose: Realtime Multi-Person 2D Pose Estimation Using Part Affinity Field," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 1, pp. 172-186, Jan. 2021, doi: 10.1109/TPAMI.2019.2929257.

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

[10] E. Insafutdinov, L. Pishchulin, B. Andres, M. Andriluka, and B. Schiele, "Deepercut: a deeper, stronger, and faster multi-person pose estimation model," in *Computer Vision – ECCV 2016,*, Lecture Notes in Computer Science, vol. 9910, Springer, Cham, Switzerland, 2016, pp. 34-50. https://doi.org/10.1007/978-3-319-46466-4_3.

[11] H.-D. Duan, J. Wang, K. Chen and D. Lin, "PYSKL: Towards Good Practices for Skeleton Action Recognition," arXiv:2205.09443v1[cs.CV], 15 May 2022, https://arxiv.org/abs/2205.09443v1 (accessed on 15.07.2022).

[12] [Online], "Papers with code. Action recognition in videos," https://paperswithcode.com/task/action-recognition-in-videos, (accessed on 15.07.2022).

[13] R. A. Jacobs, M. I. Jordan, S. J. Nowlan, and G. E. Hinton, "Adaptive mixtures of local experts", *Neural Computation*, vol. 3, no. 1, pp. 79–87, March 1991, doi: 10.1162/neco.1991.3.1.79.

[14] 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 (accessed on 15.07.2022).

[15] 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, no. 3, pp. 163–176, 2008, doi: 10.1007/s11554-008-0073-1.

[16] 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.

[17] 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, pp. 51–61, 15 June 2012, doi: 10.1016/j.neucom.2012.02.002.

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

[19] J. Liu, A. Shahroudy, D. Xu, and G. Wang, ”Spatio-Temporal LSTM with Trust Gates for 3D Human Action Recognitio,” in *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.

[20] 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.

[21] 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.

[22] 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).

[23] M. Li, S. Chen, X. Chen, Y. Zhang, Y. Wang, and Q. Tian, ”Actional-Structural Graph Convolutional Networks for Skeleton-Based Action Recognition,” in *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.

[24] 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, https://arxiv.org/abs/1805.07694v3, (accessed on 15.07.2022).

[25] 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, pp. 9532-9545, October 2020, doi: 10.1109/TIP.2020.3028207 .

[26] 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).

[27] 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.

[28] [Online], ”openpose”, CMU-Perceptual-Computing-Lab, 2021 https://github.com/CMU-Perceptual-Computing-Lab/openpose/ , (accessed on 15.07.2022).

[29] [Online], ”Keras: the Python deep learning API,” https://keras.io/ , (accessed on 15.07.2022).

[30] [Online], ”Keras Tuner,” https://keras-team.github.io/keras-tuner/ , (accessed on 15.07.2022).

[31] T. Yu and H. Zhu, ”Hyper-Parameter Optimization: A Review of Algorithms and Applications,” arXiv:2003.05689 [cs.LG], 12 Mar 2020, https://arxiv.org/abs/2003.05689 , (accessed on 15.07.2022).

[32] 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)*, vol. 42, no. 6, pp. 1453–1467, 1 June 2020, doi: 10.1109/T-PAMI.2019.2898954.

[33] J. Liu, G. Wang, P. Hu, L.-Y. Duan, and A. C. Kot, ”Global Context-Aware Attention LSTM Networks for 3D Action Recognition,” in *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.

[34] 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)*, vol. 27, no. 4, pp. 1586-1599, April 2018, doi: 10.1109/TIP.2017.2785279.