CNN-based and DTW features for human activity recognition on depth maps

Jacek Trelinski¹ · Bogdan Kwolek¹

Received: 11 August 2020 / Accepted: 28 April 2021 / Published online: 12 May 2021
© The Author(s) 2021

Abstract

In this work, we present a new algorithm for human action recognition on raw depth maps. At the beginning, for each class we train a separate one-against-all convolutional neural network (CNN) to extract class-specific features representing person shape. Each class-specific, multivariate time-series is processed by a Siamese multichannel 1D CNN or a multichannel 1D CNN to determine features representing actions. Afterwards, for the nonzero pixels representing the person shape in each depth map we calculate statistical features. On multivariate time-series of such features we determine Dynamic Time Warping (DTW) features. They are determined on the basis of DTW distances between all training time-series. Finally, each class-specific feature vector is concatenated with the DTW feature vector. For each action category we train a multiclass classifier, which predicts probability distribution of class labels. From pool of such classifiers we select a number of classifiers such that an ensemble built on them achieves the best classification accuracy. Action recognition is performed by a soft voting ensemble that averages distributions calculated by such classifiers with the largest discriminative power. We demonstrate experimentally that on MSR-Action3D and UTD-MHAD datasets the proposed algorithm attains promising results and outperforms several state-of-the-art depth-based algorithms.

Keywords Convolutional neural networks · Multivariate time-series · Ensembles · Depth-based human action recognition

1 Introduction

Automatic recognition of human activities—commonly referred to as Human Activity Recognition (HAR)—has become increasingly important task in computer vision and robotics research communities. It has become even more important after release of affordable RGB-D cameras and software for human tracking and pose estimation. The goal of visual activity recognition is to determine whether a given action occurred in image sequence. HAR has emerged as a key area of research for user-friendly interactions between humans, computers, and robots [1]. One of the goals of activity recognition is to provide information on users' behavior and intentions that can permit intelligent systems to assist users proactively with their daily and

routine tasks. Human activity recognition is a challenging task since it faces many difficulties such as high complexity of human actions, complex motion patterns, variation in motion patterns, occlusions, view point variations, illumination variations, background differences and clutter. One of the central issues is that the same action can be performed in many different ways, even by the same person. Thus, the major research challenge is to develop a proper representation of human actions, which will be both discriminative and general. According to utilized data sources, approaches to action recognition can be divided into visual sensor-based, non-visual sensor-based, and multi-modal categories. Sensor-based activity recognition is a complicated task because of inherent noisy nature of measurements or observations. Vision sensors deliver 2D images or 3D maps, whereas non-visual sensors deliver one-dimensional/multi-channel time-series data/signals. Thanks to ability to cover the subject and the context in which the activity took place, vision-based activity recognition has been at the forefront in this research area [1]. It has been a research focus for a long period of time due to

✉ Bogdan Kwolek
bkw@agh.edu.pl

¹ Department of Computer Science, AGH University of Science and Technology, 30 Mickiewicza Av., Building D-17, 30-059 Kraków, Poland

unobtrusiveness, big potential in surveillance as well as in sport. Wearable sensor-based approaches, on the other hand, do not suffer from occlusion, lighting conditions and other limitations significant for vision systems.

Although substantial progress has been made in recent years in research on behavior identification and understanding, and further progress can be expected in the near future, its associated tasks are still far from being solved and many unsolved problems remain. Because of non-rigid human shape, viewpoint variations, intra-class variations, occlusions, and plenty relevant difficulties and environmental complexities, state-of-the-art algorithms have poor performance as compared to human capabilities to recognize and interpret human motions and actions. To stimulate development and facilitate evaluation of new algorithms, several benchmark datasets have been recorded and made publicly available in the last years [2, 3]. The release of depth sensors, e.g., MS Kinect, Asus Xtion and Intel RealSense permits us to acquire information about 3D structure of scenes and objects. These sensors provide in addition to RGB image streams the depth map streams, which allow coping with viewpoints as well as illumination and color changes. This empowers the computer vision and artificial intelligence communities to move towards 3D vision, such as 3D object recognition and 3D activity analysis.

The MSR-Action3D [2] is one of the most often utilized datasets in research as well as in evaluation of algorithms relying on 3D information. The UTD-MHAD dataset [3], which has been introduced later, has four types of data modalities: RGB, depth, skeleton joint positions, and the inertial sensor signals. Most algorithms for depth-based action recognition are based on 3D positions of body joints, which can be determined, for instance, by the MS Kinect sensor [4]. However, as pointed out in a recent work [5], there are only few papers devoted to depth-based human action recognition using convolutional neural networks (CNNs). One of the reasons is that unlike RGB video-based activity analysis, 3D action recognition suffers from the lack of large-scale benchmark datasets. Moreover, the recognition performances that are achieved by such algorithms are generally lower than performances achieved by skeleton-based methods. Nevertheless, at present the choice of sensors that estimate 3D locations of body joints with sufficient accuracy is quite limited.

In the past years, traditional pattern recognition methods permitted us to attain remarkable progress in depth-based human action recognition [6, 7]. However, because these methods rely largely on predefined engineered feature with explicit parameters based on expert knowledge their generalization capabilities and recognition efficiencies are not as high as are required by real applications. In contrast to traditional handcrafted-feature-based methods, neural

network-based methods employ learnable feature extractors and computational models with multiple processing layers for action representation and recognition. One of their advantages is that HAR can be achieved through end-to-end learning. However, a huge amount of image or depth map sequences for training such models is needed, in particular for training deep neural networks. As labeling such massive collections of samples is immensely laborious and time-consuming task, the existing datasets for evaluation of performance of 3D action recognition typically have 10, 20, 27 or a little more types of actions, usually performed by a dozen or few dozen actors. Having on regard that the number of sequences in currently available datasets with the 3D data is typically smaller than one thousand, recognition of actions on the basis of only 3D depth maps is very challenging.

In this work, we present a new approach to action recognition on raw depth maps. We demonstrate experimentally that despite the limited number of training data, i.e., action sequences, it is possible to diminish overfitting/underfitting and to learn features with highly discriminative power. At the beginning, for each class we train on individual frames a separate one-against-all convolutional neural network (CNN) to extract class-specific features representing person shape. Each class-specific, multivariate time-series is processed by a Siamese multichannel 1D CNN or a multichannel 1D CNN to determine features representing actions. Optionally, for each class-specific, multivariate time-series we determine statistical features of time-series. Afterwards, for the nonzero pixels representing the person shape in each depth map we calculate handcrafted features. On multivariate time-series of such features we determine Dynamic Time Warping (DTW) features. They are determined on the basis of DTW distances between all training time-series. Finally, each class-specific feature vector is concatenated with the DTW feature vector. For each action we train a multiclass classifier that predicts probability distribution of class labels. From a pool of such classifiers we select a number of classifiers such that an ensemble built on them attains the best classification accuracy. Action recognition is achieved by a soft voting ensemble, which averages distributions calculated by such classifiers.

We demonstrate that this new algorithm has a remarkable potential. We demonstrate experimentally that on MSR-Action3D dataset the proposed algorithm outperforms state-of-the-art depth-based algorithms and attains promising results on challenging UTD-MHAD dataset. One of the most important features of our algorithm is that it needs no skeleton detection. We demonstrate experimentally that despite not utilizing the skeleton modality, the proposed algorithm attains better classification performance than several skeleton-based algorithms, which

usually achieve better results in comparison to algorithms based only on raw depth maps. It is worth noting that several digital cameras, including most stereovision ones delivers no skeleton modality. Moreover, the operative measurement range of stereovision cameras can be far longer in comparison to depth range of operation of structured light/ToF sensors.

2 The algorithm

In Sect. 2.1 we present class-specific features, whereas DTW features for action representation are discussed in Sect. 2.2. The learned frame-features and descriptors of time-series of class-specific frame-features are outlined in Sects. 2.1.1 and 2.1.2, respectively. Afterwards, statistical frame-features and DTW-based features describing time-series of such statistical features are discussed in Sects. 2.2.1 and 2.2.2, respectively. In next two Subsections we describe the ensemble for action classification.

2.1 Action descriptors based on class-specific features

2.1.1 Learned frame-features

Having on regard that the datasets for depth-based action recognition have insufficient number of sequences to learn deep models with adequate generalization capabilities, we utilize CNNs operating on single depth maps for extraction of features representing person shape [8]. In the discussed approach a separate CNN is trained for each action class to predict if the considered frame belongs to the class for which the CNN had been trained or to one of the remaining classes, same as in one-vs-all (OvA) also referred to as one-vs-rest (OvR) multiclass classification. Each network is trained on single frames belonging to the considered class and frames sampled from pool of images from remaining classes. The trained networks are then used to extract features representing person shapes in raw depth maps. One hundred of features are extracted on the basis of the penultimate layer of the CNN. Since the number of frames in depth maps sequences is not identical, the lengths of multivariate time-series representing actions can differ in length.

In this work, the input rectangular windows encompassing persons in the raw depth maps are scaled to sizes 64×64 pixels. The input of convolutional neural network is a $3 \times 64 \times 64$ tensor consisting of frontal depth map, and orthogonal projection of the input depth map onto xz and yz planes. This indicates that we determine side-view and top projections of depth maps. Specifically, the input depth map is projected onto three 2D orthogonal Cartesian

planes, where xy plane represents frontal view, yz plane exemplifies the side view and xz plane represents top view [3]. The architecture of neural network is depicted on Fig. 1. The output of the CNN is a softmax layer with number of neurons equal to the number of action categories to be recognized. Each neural network has been trained in 200 epochs, batch size set to 32, using Nesterov momentum with learning rate set to 0.001 and momentum equal to 0.9. After the training, the outputs of the hidden layers have been employed to extract features representing person's shape. A depth maps sequence representing an action is described by a multivariate time-series of length equal to number of frames and dimension equal to one hundred.

2.1.2 Temporal descriptors of time-series of class-specific frame-features

Embedding action features using multichannel 1D CNN

In multi-channel, temporal CNNs (1D CNNs) the 1D convolutions are applied in the temporal dimension. In this work, time-series (TS) of frame-features that were extracted by the 2D convolutional neural network have been used to train multi-channel 1D CNNs. The 1D CNNs have then been used for embedding the action features. The number of channels is equal to 100, see also Fig. 5. The multivariate time-series were interpolated to length equal to 64. Each 1D CNN is trained to classify all actions in the considered dataset. It operates on class-specific features, which are discussed in Sect. 2.1.1. Cubic-spline algorithm has been utilized to interpolate the TS to such a common length. The first layer of the 1D CNN is a filter (feature detector) operating in time dimension. Having on regard that the amount of the training data in current datasets for depth-based action recognition is quite small, we utilize a shallow neural network that consists of two convolutional layers, each with 8×1 filter, 4×1 and 2×1 max pools, respectively, see Fig. 2. The number of neurons in the dense layer is equal to 100. For each multivariate time-series of CNN-based frame-features a separate multichannel 1D CNN has been trained. Such approach is due to redundant depth maps, i.e., the same human poses in different actions. The number of output neurons is equal to number of the classes. Nesterov Accelerated Gradient (Nesterov momentum) has been used to train the network in 1000 iterations, with momentum set to 0.9, dropout equal to 0.5, learning rate equal to 0.001, L1 parameter set to 0.001. After training the 1D CNNs, the outputs of the dense layers have been used to extract the features. An implementation of 1D CNN available at Keras repository¹ has been used for embedding the discussed class-specific action features.

¹ https://keras.io/api/layers/convolution_layers/convolution1d/.

Fig. 1 Flowchart of the CNN for learning class-specific frame-features

Fig. 2 Multichannel 1D CNN for embedding class-specific action features

Embedding action features using Siamese neural network

Siamese neural network is a special kind of network that shares weights while working in tandem on two different input vectors to compute similarity of the inputs by extracting and comparing feature vectors. The feature vectors can be considered as the semantic similarity between the projected representation of the two input vectors. A Siamese neural network (SNN) sometimes called a twin neural network has been introduced in [9]. SNNs have been utilized for dimensionality reduction [10] as well for one-shot image classification [11]. Siamese networks perform well in one-shot approaches since their shared weights mean there are less parameters to learn during the training. Thus, this architecture can produce good results with a relatively small amount of training data. It can be used to determine if two exemplars are of the same class. The SNN does this by reducing the data

dimensionality and employing a distance-based cost function to differentiate between the classes.

The central idea behind SNNs is to learn an embedding such similar data pairs are close to each other and dissimilar data pairs are separated by a distance that depends on a parameter called margin. The Siamese neural network reduces the dimensionality by mapping input vectors with the same class to nearby points in a low-dimensional space. It is trained on paired data (x_p, x_q, y_{pq}) , where the distance between a pair of examples with the same label is minimized, whereas distinct-class pairs are penalized for being smaller than the margin m , and the label $y_{pq} \in \{0, 1\}$ indicates whether a pair (x_p, x_q) is from the same class or not. The contrastive loss function L can be expressed as follows:

$$L(\theta, x_p, x_q) = y_{pq} \|f(x_p) - f(x_q)\|_2^2 + (1 - y_{pq}) \{\max(0, m - \|f(x_p) - f(x_q)\|_2)\}^2 \quad (1)$$

where function $f(\cdot)$ stands for the feature embedding from the network. The loss function expresses how well the function $f(\cdot)$ is capable of placing similar image representations in the close proximity and keep dissimilar image representations distant. It penalizes positive pairs by the squared Euclidean distances and negative pairs by the squared difference between the margin m and the Euclidean distance for pairs with distance less than the margin m .

In the proposed approach the Siamese neural network operates on time-series of class-specific frame-features. Figure 3 depicts diagram of the utilized Siamese neural network. The size of input data is 64×100 , i.e., length of interpolated time-series times the number of channels. In this multi-channel, temporal Siamese network the 1D convolutions are applied in the temporal dimension. The Siamese neural networks have been trained in 300 epochs, with batch size set to 32, L1 regularization equal to 0.01, using Adam optimizer with learning rate set to 0.00006.

2.1.3 Statistical features of time-series

For each multivariate time-series of CNN-based frame-features, which can differ in size due to different number of frames in depth maps sequences we calculate statistical features. Such statistical features represent actions. For each time-series feature we calculate four features: average, standard deviation, skewness and correlation of the time-series with time. The resulting features are called statistical temporal features (STF). The motivation for using skewness was to include information about asymmetry in random variable's probability distribution [8]. The size of STF representing time-series of CNN-based frame-features is equal to 400 (4×100).

2.2 DTW-based features for action representation

2.2.1 Statistical frame-features

For each depth frame we calculate also statistical features describing the person's shape. Similarly to learned frame-features that have been described in Sect. 2.1.1, we project the acquired depth maps onto three orthogonal Cartesian views to capture the 3D shape and motion information of human actions. Only pixels representing the extracted person in depth maps are utilized for calculating such features. The following vectors of frame-features were calculated on such depth maps:

1. Standard deviation (axes x , y , z),
2. Skewness (axes x , y , z),
3. Correlation (xy , xz and zy axes),

4. x -coordinate for which the corresponding depth value represents the closest pixel to the camera, y -coordinate for which the corresponding depth value represents the closest pixel to the camera.

This means that the person shape in each depth maps is described by 3, 3, 3, and 2 features, respectively, depending on the chosen feature set. A human action represented by a number of depth maps is described by a multivariate time-series of length equal to number of frames and dimension 2 or 3 in dependence on the chosen feature set.

2.2.2 DTW-based features

Dynamic time warping (DTW) is an effective algorithm for measuring similarity between two temporal sequences, which may vary in speed and length. It calculates an optimal match between two given sequences, e.g., time series [12]. One of the most effective algorithms for time-series classification is 1-NN-DTW, which is a special k -nearest neighbor classifier with $k = 1$ and a dynamic time warping for distance measurement. In DTW the sequences are warped nonlinearly in time dimension to determine the best match between two samples such that when the same pattern exists in both sequences, the distance is smaller. Let us denote $D(i, j)$ as the DTW distance between sub-sequences $x[1 : j]$ and $y[1 : j]$. Then the DTW distance between x and y can be determined by the dynamic programming algorithm according to the following iterative equation:

$$D(i, j) = \min\{D(i-1, j-1), D(i-1, j), D(i, j-1)\} + |x_i, y_j| \quad (2)$$

The time complexity of calculation of DTW distance is $O(nm)$, where n and m are the length of x and y , respectively.

We calculate the DTW distances between all depth maps sequences in the training subset. For each depth map sequence the DTW distances were calculated for the features sets 1, 2, 3 and 4. The DTW distances were then used as features. This means that the resulting feature vector has size $n_t \times 4$, where n_t denotes the number of training depth map sequences. The algorithms used for calculating the DTW are implemented in the Python modules DTADistance [13].

2.3 Multiclass classifiers to construct ensemble

The features described in Sects. 2.1 and 2.2 were used to train multiclass classifiers representing probability distribution of class labels, see Fig. 4 that depicts a flowchart of a single ensemble classifier. Having on regard that for each class in order to extract depth map features a class-specific

Fig. 3 Schematic diagram of Siamese multichannel 1D CNN operating on class-specific frame-features

classifier has been trained, the number of such classifiers is equal to the number of classes to be recognized. The CNNs operating on depth maps (Sect. 2.1.1) deliver time-series of CNN-based frame features, on which we determine Siamese features, 1D CNN-based features or optionally statistical temporal features (Sect. 2.1.2). The DTW-based features (Sect. 2.2.2) representing actions, c.f. right branch on Fig. 4, are concatenated with the mentioned above action features, i.e., embeddings of class-specific frame-features that are calculated by Siamese neural networks or multichannel 1D CNNs, or optionally they are concatenated with statistical features on time-series of class-specific frame-features, c.f. left branch on Fig. 4. The most informative DTW-based features were selected using recursive feature selection (RFE) algorithm. The multiclass classifiers generating the probability distribution of class labels were finally used to construct an ensemble responsible for classification of actions.

2.4 Ensemble of classifiers

Figure 5 depicts the ensemble for action classification. The final decision of the ensemble is calculated on the basis of voting of classifiers, which are depicted on Fig. 4. The multiclass classifier operates on concatenated Siamese features, 1D CNN features or optionally STF features (class-specific branch) with the selected DTW-based features (DTW branch), see Fig. 5. Four feature vectors determined by the DTW, see also Fig. 4 and four channels in the block labeled as DTW features, are concatenated and then the concatenated vector is fed to the feature selector. As we can see, the class-specific Siamese features, 1D CNN features or optionally STF features are concatenated with CFV (common feature vector), and then used to train multiclass classifiers.

Optionally, we perform classifier selection. We generate 1000 sets of classifiers, where each set is created by

Fig. 4 Multi-class classifier to construct ensemble

selecting without replacement k classifiers from all K classifiers, k is sampled randomly from integer numbers in range $[2, \dots, K]$, and K denotes the number of classes. The selection of the best classifiers is done on the validation subset that is identical for all sets of classifiers. For each classifier set from 1000 sets of classifiers we calculate the accuracy achieved by the ensemble consisting of the considered classifiers. The set of classifiers, which attains the best classification accuracy on the common validation subset is selected to build the final ensemble.

3 Experimental results and discussion

The proposed algorithm has been evaluated on two publicly available benchmark datasets: MSR Action3D dataset [2] and UTD-MHAD dataset [3]. The datasets were selected having on regard their frequent use by action recognition community in the evaluations and algorithm comparisons. In all experiments and evaluations, 557 sequences of MSR Action3D dataset were investigated.

Half of the subjects were utilized to provide the training data and the rest of the subjects have been employed to get the test subset. In the discussed classification setting, all classes were considered in a single setting, which is different from evaluation protocols based on AS1, AS2 and AS3 data splits in which accuracies are calculated in every data split and then averaged. Another aspect of this is that the classification performances achieved in the utilized setting are lower in comparison to classification performances, which are achieved on AS1, AS2, AS3 setting due considering all actions in one setting instead of considering similar actions in three groups of actions. The cross-subject evaluation protocol [7, 14] has been applied in all evaluations. Specifically, in the cross-subject protocol, odd subjects are used for training (1, 3, 5, 7, and 9) and even subjects (2, 4, 6, 8, and 10) are employed for testing. The discussed evaluation protocol is different from the evaluation procedure employed in [15], in which more subjects were utilized in the training subset.

The UTD-MHAD dataset [3] comprises twenty seven different actions performed by eight subjects (four females

Fig. 5 Ensemble operating on handcrafted features concatenated with class-specific features

and four males). Every performer repeated each action four times. All actions were performed in an indoor environment with fixed background. The dataset was collected using the Kinect sensor and a wearable inertial sensor. It consists of 861 data sequences. The evaluation protocol used for this dataset follows the cross-subject protocol, where odd subjects were used for training and even subjects for testing, same as settings in [3].

At the beginning of experiments we extracted 11,132 depth maps from the MSR Action3D as well as 29,199

depth maps from the MHAD dataset in order to train neural networks responsible for extracting class-specific frame features. Given the trained neural networks we determined 56,700 time-series for the MSR Action3D dataset and 86,100 time-series for the UTD MHAD dataset. In order to confirm the legitimacy of use of the skewness we calculated p values of Shapiro-Wilk test. For each time-series through checking if p values are less than 0.05 (for a 95% confidence interval) the null hypothesis that the time-series come from a normal distribution were rejected. For the

MSR Action3D dataset the percentage of rejected hypotheses was equal to 40.3%, whereas for the UTD MHAD dataset the percentage of rejected hypotheses was equal to 47.6%. We executed also robust Jarque–Bera test [16]. For the MSR Action3D dataset the percentage of rejected hypotheses was equal to 22.5%, whereas for the UTD MHAD dataset the percentage of rejected hypotheses was equal to 27.5%. Significant percentage of time-series departing from normality justified the use of the proposed STF features. Afterwards, we trained neural networks discussed in Sects. 2.1.1 and 2.1.2, trained the classifiers and then built the ensembles.

Table 1 presents experimental results that were achieved on the MSR Action 3D dataset. Comparing results in rows A2 with A1 and A3, results in rows C2 with C1 and C3, and then D2 with D1 and D3, and finally E2 with E1 and E3 we can observe that statistical features permit to achieve better results in comparison to 1D_CNN as well as Siamese (sim) features. The DTW features allow to achieve far better recognition performance in comparison to the stats features, 1D_CNN features as well as sim features, compare results in row B with results in rows A1, A2, and A3. Combining DTW features with the sim features, 1D_CNN features or stats features leads to considerable gains in the recognition performance. However, not all DTW features have high discriminative power. Thus, the Recursive Feature Elimination (RFE) leads to further improvements of the classification accuracies, see results in rows D1–D3. The ensemble built on the selected classifiers from a pool of twenty classifiers as well as DTW features selected by the RFE permits achieving further gains in the classification accuracies, cf. results in rows

E1–E3. The best results were achieved by the ensemble built on class-specific classifiers operating on stats features, which were concatenated with the DTW features acting as the common features, cf. results in row E2. Figure 6 depicts the confusion matrix, which has been obtained by the proposed algorithm. In a configuration in which the algorithm was built using 1D_CNN features instead of the stats features, only four class-specific classifiers would be needed to achieve the best results, cf. results in row E1.

Table 2 illustrates the classification performance of the proposed method on the MSR-Action3D dataset in comparison to previous depth-based methods. The classification performance has been determined using the cross-subject evaluation [17], where subjects 1, 3, 5, 7, and 9 were utilized for training and subjects 2, 4, 6, 8, and 10 were employed for testing. The proposed method achieves better classification accuracy in comparison to recently proposed method [18] (Split II), and it has worse performance than recently proposed methods [14, 19]. This can be explained by limited amount of training samples in the MSR-Action3D dataset. To deal with this, Wang et al. synthesized training samples on the basis of 3D points. In consequence, the discussed algorithm is not based only on depth maps. Results attained by our best performing algorithm are better than results achieved by recently proposed Action-fusion method [20]. It is also worth noting that algorithm built on the sim features slightly outperforms the algorithm mentioned above.

Table 3 presents experimental results that were achieved on the UTD-MHAD dataset. Comparing the results in the first three rows we can notice that 1D_CNN features permit to achieve better results in comparison both to the stats

Table 1 Recognition performance on MSR Action 3D dataset

Common feat. vect.	Class-specific	num. DTW feat.	Num. deep feat.	Num. cl.	Accuracy	Precision	Recall	F1-score
[A1] _	1D_CNN	0	100	20	0.7273	0.7317	0.7273	0.6926
[A2] _	Stats	0	400	20	0.7818	0.8047	0.7818	0.7621
[A3] _	Sim	0	100	20	0.7345	0.7265	0.7345	0.7120
[B] DTW	–	1168	0	1	0.8764	0.8857	0.8764	0.8700
[C1] DTW	1D_CNN	1168	100	20	0.9091	0.9173	0.9091	0.9050
[C2] DTW	Stats	1168	400	20	0.9164	0.9243	0.9164	0.9131
[C3] DTW	Sim	1168	100	20	0.9091	0.9139	0.9091	0.9034
[D1] DTW, RFE	1D_CNN	350	100	20	0.9091	0.9129	0.9091	0.9059
[D2] DTW, RFE	Stats	350	400	20	0.9309	0.9370	0.9309	0.9292
[D3] DTW, RFE	Sim	350	100	20	0.9200	0.9204	0.9200	0.9156
[E1] DTW, RFE	1D_CNN	350	100	4	0.9309	0.9323	0.9309	0.9278
[E2] DTW, RFE	Stats	350	400	6	0.9564	0.9594	0.9564	0.9542
[E3] DTW, RFE	Sim	350	100	8	0.9455	0.9471	0.9455	0.9421

Bold values indicate the best results

Fig. 6 Confusion matrix on MSR-Action3D dataset, obtained by the ensemble

Table 2 Comparative recognition performance of the proposed method with recent algorithms on MSR Action 3D dataset

Method	Split	Modality	Acc. (%)
3DCNN [21]	Split II	Depth	84.07
DMMs [6]	Split II	Depth	88.73
PRNN [18]	Split II	Depth	94.90
WHDMM + CNN [14]	Split I	Depth + syntch.	100.00
S DDI [19]	Split I	Depth + syntch.	100.00
3DHoT-MBC [22]	Split I	Depth + RGB	95.24
Action-fusion [20]	Split I	Depth	94.51
Proposed method	Split I	Depth	95.64

Bold values indicate the best results

features and the sim features. The DTW features permit to attain better results in comparison to features discussed above, cf. results in row B and rows B2-B3, with the exception of the 1D_CNN features, cf. results in row B and row B1. Combining DTW features with the 1D_CNN features, or the stats features, or the sim features leads to considerable gains in the recognition performance. The DTW features selected by RFE and combined with the 1D_CNN features, or the stats features, or the sim 1D_CNN features lead to better results in comparison to results of algorithm with no RFE. The best classification accuracies were achieved by an algorithm built on class-specific classifiers operating on 1D_CNN features, which were concatenated the with DTW features acting as the common features. As we can observe, the best classification performance was achieved by the ensemble built on seven classifiers that were selected from a pool of twenty

seven classifiers as well as DTW features selected by the RFE. Feature selection together with classifier selection leads to far better results. In a configuration in which the algorithm was built using the sim features instead of the 1D_CNN features, only five class-specific classifiers would be needed to achieve the best results, cf. results in the last row. The ensemble built on the sim features, which were concatenated with the DTW features acting as the common features has been ranked as the second place taking into account the classification performance on both datasets. By comparing results from Tables 2 and 4 we can observe that although the WHDMM algorithm that rely not only on raw depth maps achieved better classification performance than our algorithm, it achieves worse results on the UTD-MHAD dataset in comparison to results achieved by the proposed algorithm. Figure 7 depicts the confusion matrix corresponding to best results from Table 3, which has been obtained by the proposed algorithm.

Table 4 presents the recognition performance of the proposed method compared with previous methods. Most of current methods for action recognition on UTD-MHAD dataset are based on 3D positions of body joints. These methods usually achieve better results than methods relying on depth data only. Despite that our method employs only depth modality, it outperforms many of them. Methods based on depth data only have wider range of applications since not all depth cameras have support for skeleton extraction. Our method noticeably outperforms the WHDMM + 3DConvNets method that utilizes weighted hierarchical depth motion maps (WHDMMs) and three 3D ConvNets. The WHDMMs are applied at several temporal scales to encode spatiotemporal motion patterns of actions into 2D spatial structures. In order to collect sufficient amount of training data, the 3D points are rotated and then utilized to synthesize new exemplars. In contrast, our method operates only on raw depth maps. Results achieved by our algorithm are identical with results achieved by recently proposed Action-fusion method [20]. On both datasets our method outperforms 3DHoT-MBC algorithm [22], which is an ensemble of base classifiers on different types of features encoding motion information across depth frames and local texture variation simultaneously.

Like relevant action recognition systems, our algorithm also requires tuning a few parameters so as to achieve improvements in classification performance with respect to a performance with default settings. As can be seen in Tables 1 and 3 the best classification performances are achieved by configurations denoted as E1--E3. The best results were obtained through selecting 350 features, though 300 features was the best choice for one of the configurations. The sim features permitted to achieve classification performances that have been ranked as the

Table 3 Recognition performance on UTD-MHAD dataset

Common feat. vect.	Class-specific	Num. DTW feat.	Num. deep feat.	Num. cl.	Accuracy	Precision	Recall	F1-score
[A1] _	1D_CNN	0	100	27	0.7907	0.8119	0.7907	0.7843
[A2] _	Stats	0	400	27	0.7302	0.7354	0.7302	0.7153
[A3] _	Sim	0	100	27	0.7767	0.8021	0.7767	0.7740
[B] DTW	–	1724	0	1	0.7860	0.8146	0.7860	0.7842
[C1] DTW	1D_CNN	1724	100	27	0.8628	0.8772	0.8628	0.8614
[C2] DTW	Stats	1724	400	27	0.8326	0.8461	0.8326	0.8301
[C3] DTW	Sim	1724	100	27	0.8339	0.8349	0.8552	0.8349
[D1] DTW, RFE	1D_CNN	300	100	27	0.8698	0.8842	0.8698	0.8694
[D2] DTW, RFE	Stats	350	400	27	0.8419	0.8617	0.8419	0.8386
[D3] DTW, RFE	Sim	300	100	27	0.8349	0.8580	0.8349	0.8347
[E1] DTW, RFE	1D_CNN	300	100	7	0.8814	0.8979	0.8814	0.8804
[E2] DTW, RFE	Stats	350	400	13	0.8605	0.8716	0.8605	0.8578
[E3] DTW, RFE	Sim	300	100	5	0.8744	0.8935	0.8744	0.8766

Bold values indicate the best results

Table 4 Comparative recognition performance of the proposed method with recent algorithms on MHAD dataset

Method	Modality	Accuracy (%)
JTM [21]	Skeleton	85.81
SOS [18]	skeleton	86.97
Kinect and inertial [3]	Skeleton	79.10
Struct. body [19]	Skeleton	66.05
Struct. part [19]	Skeleton	78.70
Struct. joint [19]	Skeleton	86.81
Struct. S DDI [19]	Skeleton	89.04
3DHoT-MBC [22]	Depth + RGB	84.40
WHDMMs + ConvNets [14, 19]	Depth	73.95
Action-fusion [20]	Depth	88.14
Proposed method	Depth	88.14

Bold values indicate the best results

second place across the E1–E3 performances at both datasets. The 1D_CNN features, which permitted to achieve best results on UTD MHAD dataset, gave slightly worse results considering the E1–E3 classification performances. The selection of class-specific classifiers permits to achieve improvements of classification performance as well as simplification of the final ensemble. We experimented with various configurations of the ensemble, with different optimizers for training neural networks, activation functions, but it turned out that the presented configurations are the most stable. Since DTW has high computational complexity, the classification time depends mainly on it. The utilized DTAIDistance implementation [13] permits calculation of DTW distances between all sequences in a list of sequences as well as

Fig. 7 Confusion matrix on UTD-MHAD dataset, obtained by the ensemble

speeding up the computations by calling and executing the code implemented in C. Approximate DTW algorithms that provide optimal or near-optimal alignments with an $O(N)$ time and memory complexity can also be used in our algorithm. In order to include high-level information, for instance extracted on the basis of point clouds through 3D object detection, and at the same time to reduce retraining of models [23] we are going to employ fuzzy rules, similar to approach in [24], where the fuzzy rules work as a classification ensemble.

4 Conclusions

In this paper, we presented a new algorithm for human action recognition on raw depth maps. At the beginning, for each class we train a separate one-against-all convolutional neural network to extract class-specific features representing person shapes in raw depth maps. For each class-specific, multivariate time-series we determine multichannel 1D CNN features or Siamese features. Alternatively, instead of the mentioned above features we determine statistical features of time-series. Afterwards, for the nonzero pixels representing the person shape in each raw depth map we calculate handcrafted features. On multivariate time-series of such features we determine Dynamic Time Warping features. They are determined on the basis of DTW distances between all training time-series. Finally, each class-specific feature vector is concatenated with the DTW feature vector. For each action we train a multiclass classifier that predicts probability distribution of class labels. From ensemble of classifiers we select a number of classifiers with largest number of correctly predicted classes. Action recognition is performed by a soft voting ensemble, which averages class distributions determined by selected best classifiers. We demonstrated experimentally that the proposed class-specific features and DTW-based features have considerable discriminative power. The concatenation of the mentioned above features leads to remarkable gains in classification performance. The proposed ensemble that averages probability distributions determined by individual classifiers permits achieving better results, particularly if it is built on classifiers with the best classification scores. We demonstrated experimentally that on MSR-Action3D and UTD-MHAD datasets the proposed algorithm attains promising results and outperforms several state-of-the-art depth-based algorithms. Our algorithm outperforms most recent skeleton-based methods, which typically achieve better results than methods based on depth maps only. The source code of the presented algorithm is available at <https://github.com/tjacek/ActionClassifier>.

Acknowledgements This work was supported by Polish National Science Center (NCN) under a research Grant 2017/27/B/ST6/01743.

Declarations

Conflict of interest We would like to state that there is no conflict of interest to declare and corresponding author shall bear full responsibility for the submission.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate

if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Liang B, Zheng L (2015) A survey on human action recognition using depth sensors. In: International conference on digital image computing: techniques and applications, pp 1–8
2. Li W, Zhang Z, Liu Z (2010) Action recognition based on a bag of 3D points. In: IEEE international conference on computer vision and pattern recognition—workshops, pp 9–14
3. Chen C, Jafari R, Kehtarnavaz N (2015) UTD-MHAD: a multimodal dataset for human action recognition utilizing a depth camera and a wearable inertial sensor. In: IEEE international conference on image processing (ICIP), pp 168–172
4. Shotton J, Sharp T, Kipman A, Fitzgibbon A, Finocchio M, Blake A, Cook M, Moore R (2013) Real-time human pose recognition in parts from single depth images. *Commun ACM* 56(1):116–124
5. Wu H, Ma X, Li Y (2019) Hierarchical dynamic depth projected difference images-based action recognition in videos with convolutional neural networks. *Int J Adv Robot Syst*. <https://doi.org/10.1177/1729881418825093>
6. Yang X, Zhang C, Tian YL (2012) Recognizing actions using depth motion maps-based histograms of oriented gradients. In: Proceedings of the 20th ACM international conference on multimedia. ACM, pp 1057–1060
7. Xia L, Aggarwal J (2013) Spatio-temporal depth cuboid similarity feature for activity recognition using depth camera. In: IEEE international conference on computer vision and pattern recognition, pp 2834–2841
8. Trelinski J, Kwolek B (2019) Ensemble of classifiers using CNN and hand-crafted features for depth-based action recognition. In: Proceedings of international conference on artificial intelligence and soft computing. LNCS, vol 11509. Springer, pp 91–103
9. Chopra S, Hadsell R, LeCun Y (2005) Learning a similarity metric discriminatively, with application to face verification. In: CVPR, pp 539–546
10. Hadsell R, Chopra S, LeCun Y (2006) Dimensionality reduction by learning an invariant mapping. In: CVPR. IEEE Computer Society, pp 1735–1742
11. Koch G, Zemel R, Salakhutdinov R (2015) Siamese neural networks for one-shot image recognition. In: ICML deep learning workshop, vol 2
12. Paliwal K, Agarwal A, Sinha S (1982) A modification over Sakoe and Chiba's dynamic time warping algorithm for isolated word recognition. *Signal Proc* 4(4):329–333
13. Meert W, Craenendonck TV (2018) DTAIDistance. Time series distances: dynamic time warping (DTW). [Online]. Available <https://zenodo.org/record/1202379>
14. Wang P, Li W, Gao Z, Zhang J, Tang C, Ogunbona P (2016) Action recognition from depth maps using deep convolutional neural networks. *IEEE Trans Hum Mach Syst* 46(4):498–509
15. Xia L, Chen C-C, Aggarwal J (2012) View invariant human action recognition using histograms of 3D joints. In: CVPR workshops, pp 20–27
16. Stehlík M, Štěpánek L, Thulin M (2014) On robust testing for normality in chemometrics. *Chemom Intell Lab Syst* 130:98–108

17. Wu Y (2012) Mining actionlet ensemble for action recognition with depth cameras. In: IEEE international conference on computer vision and pattern recognition, pp 1290–1297
18. Hou Y, Li Z, Wang P, Li W (2018) Skeleton optical spectra-based action recognition using convolutional neural networks. *IEEE Trans Circuits Syst Video Technol* 28(3):807–811
19. Wang P, Wang S, Gao Z, Hou Y, Li W (2017) Structured images for RGB-D action recognition. In: IEEE international conference on computer vision workshops (ICCVW), pp 1005–1014
20. Kamel A, Sheng B, Yang P, Li P, Shen R, Feng D (2019) Deep convolutional neural networks for human action recognition using depth maps and postures. *IEEE Trans Syst Man Cybern Syst* 49(9):1806–1819
21. Wang P, Li W, Li C, Hou Y (2018) Action recognition based on joint trajectory maps with convolutional neural networks. *Knowl Based Syst* 158:43–53
22. Zhang B, Yang Y, Chen C, Yang L, Han J, Shao L (2017) Action recognition using 3D histograms of texture and a multi-class boosting classifier. *IEEE Trans Image Process* 26(10):4648–4660
23. de Jesus Rubio J (2009) SOFMLS: online self-organizing fuzzy modified least-squares network. *IEEE Trans Fuzzy Syst* 17(6):1296–1309
24. Korytkowski M, Scherer R, Szajerman D, Polap D, Wozniak M (2020) Efficient visual classification by fuzzy rules. In: IEEE international conference on fuzzy system (FUZZ-IEEE), pp 1–6

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.