Multi-channels CNN Temporal Features for Depth-based Action Recognition

Trelinski Jacek, Bogdan Kwolek

AGH University of Science and Technology
30 Mickiewicza Av., 30-059 Krakow, Poland

Abstract

In this paper, we investigate temporal features that are extracted by a multi-channel convolutional neural network in depth map-based human action recognition. At the beginning, for the non-zero pixels representing the person shape in each depth map we calculate handcrafted features. On multivariate time-series of such handcrafted features we train a multi-class, multi-channel CNN to model temporal features as well as we extract statistical features of time-series. The concatenated features are stored in a common feature vector. Afterwards, for each class we train a separate one-against-all convolutional neural network to extract class-specific features of depth maps. For each class-specific, multivariate time-series we calculate statistical features of time-series. Finally, each class-specific feature vector is concatenated with the common feature vector resulting in an action feature vector. For each action represented by action feature vectors we train a multi-class classifier with one-hot encoding of output labels. The recognition of the action is done by a voting-based ensemble operating on such one-hot encodings. We demonstrate experimentally that on UTD-MHAD dataset the proposed algorithm outperforms state-of-the-art depth-based algorithms and attains promising results on MSR-Action3D dataset.

1. INTRODUCTION

Human action recognition (HAR) can be defined as an ability to determine whether a given action occurred in image or depth sequence. This is a challenging problem due to high complexity of human actions, occlusions, complex motion patterns, variation of appearance, variation in motion patterns, etc. Moreover, each action can be done in many different ways. With regard to data collection process, the approaches to action recognition can be divided into visual sensor-based, non-visual sensor-based, and multi-modal categories. Visual sensors provide data in the form of 2D or 3D images, whereas other sensors provide the data in the form of one-dimensional/multi-channel signal. Visual sensor-based approaches are one of the most popular approaches due to unobtrusiveness, big potential in surveillance as well as ability to cover the subject and the context in which the activity took place [1].

Due to huge application potential, researchers devoted much effort during the past few decades to HAR. However, due to non-rigid shape of the humans, intra-class variations, viewpoint variations, occlusions and plenty another challenges and environmental complexities, the current state-of-the-art algorithms have poor performance in comparison to human ability to recognize and to understand human motions and actions. In order to stimulate the research as well as to facilitate development and evaluation of new algorithms, many benchmark datasets have been created in the last decades [2,3]. Prior to the release of low-cost depth cameras, such as Microsoft Kinect, research has concentrated on learning and recognizing actions using RGB datasets and video repositories.

When 3D objects are captured using 2D cameras then one dimension is lost during the acquisition, which causes the loss of some important information. For the same reason, 3D-based approaches provide higher accuracy than 2D-based approaches. The introduction of low-cost integrated depth sensors being able to capture both RGB and depth information has significantly advanced the research on human action recognition. The MSR-Action3D [2], UTKinect are one of the most frequently used dataset in the research as well as in evaluation of the algorithms. The recently introduced UTD-MHAD dataset [3] has four types of data modalities: RGB, depth, skeleton joint positions, and the inertial sensor signals and it is considered as a valuable benchmark data. Most of the approaches to depth-based action recognition are based on the skeleton extracted by the MS Kinect sensor. The number of approaches, which rely on only 3D maps is quite limited. The accuracies that are achieved by such algorithms are usually lower in comparison to accuracies of skeleton-based methods. However, at present the number of camera types, which estimate locations of body joints with sufficient 3D accuracy is quite limited. Moreover, the maximum range of existing depth sensors that rely on structured light or TOF technologies is usually below six meters. This limits their applications in many areas, including surveillance applications.

Traditional approaches to activity recognition rely on the handcrafted feature-based representations [4,5]. Unlike handcrafted representation-based approaches, where actions are represented by engineered features, learning-based approaches have the capability to discover the most informative features automatically from raw data. Deep learning-based methods have capability of processing the images/videos in their raw forms and automating the process of feature extraction, representation, and classification. These methods utilize trainable feature extractors and computational models with multiple processing layers for action representation and recognition. This means that human action recognition can be achieved through end-to-end learning. The deep learning-based models for human action recognition require a huge amount of image or depth map sequences for training. Collecting and annotating huge amounts of data is immensely laborious and requires suitable equipment and computational resources. The currently available datasets for 3D action recognition typically have 10, 20, 27 or a little more types of actions, which were performed by a dozen or dozen actors, and each action has been repeated a few times. Due to limited number of data sequences in the currently available datasets with the 3D data, which is typically smaller than one thousand, the recognition of action on the basis of 3D depth maps is very challenging.

In this work we present a new approach to action recognition. 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. The depth map sequences that we classify are not of a fixed length, i.e. we classify variable length time-series. At the beginning, for the non-zero pixels representing the person shape in each depth map we calculate handcrafted features of depth maps. On multivariate time-series of such handcrafted features we train a multi-class, multi-channel CNN (MC CNN) to model temporal features as well as we extract statistical features of time-series. The concatenated features are stored in a common feature vector (CFV). As far as we know, until now the multi-channel convolutional neural networks [6, 7] were not used in human action recognition. Afterwards, for each class we train a separate one-against-all convolutional neural network to extract class-specific features of depth maps. For each class-specific, multivariate time-series we calculate statistical features of time-series. Finally, each class-specific feature vector is concatenated with the common feature vector resulting in an action feature vector. For each action represented by action feature vectors we train a multi-class classifier with one-hot encoding of output labels. The recognition of the action is done by a voting-based ensemble operating on such one-hot encodings.

We demonstrate that this new algorithm, which differs in several aspects from current algorithms for human action classification has considerable potential. We demonstrate experimentally that on UTD-MHAD dataset the proposed algorithm outperforms state-of-the-art depth-based algorithms and attains promising results on MSR-Action3D dataset. One of the most important characteristic of the proposed method is that it does not require skeleton detection. We demonstrate experimentally that despite not using the skeleton, our algorithm achieves better accuracies than several skeleton-based algorithms. Owing to the use of depth maps only, our algorithm can be used on depth data provided by stereo cameras, which can deliver the depth data for persons being at larger distances to the sensors. Moreover, it is well known that the Kinect sensor fails to estimate the skeleton in several scenarios, including scenarios with human fall detection. A non-trivial use of various features, learned on different domains, like single depth map, time-sequence of depth maps, results in a statistical algorithm for action recognition, in which the final decision is done by an ensemble.

2. THE ALGORITHM

In Subsection 2.1 we present features describing the person's shape in single depth map. At he beginning of Subsection 2.1.1 we describe how features describing the person's shape in single depth map are learned. Afterwards, we describe how handcrafted frame-features are extracted. In Subsection 2.2 we discuss CNN-based time-series classification and representation. Next, we present statistical features of time-series. In next two subsections we describe the ensemble.

2.1 Frame-Features

2.1.1 Learned Frame-Features

Since current datasets for depth-based action recognition have insufficient number of sequences to learn deep models with adequate generalization capabilities, we propose to use CNNs operating on single depth maps to extract informative features. Because the number of frames in the current benchmark datasets for RGBD-based action recognition is pretty large, it is possible to train a set of CNNs on single depth maps. In the proposed approach a separate CNN is trained for each action class to distinguish between this class and all remaining classes, same as in one-vs-all multi-class classification. In other words, each CNN is trained to predict if the considered depth maps belongs to the class for which the CNN had been trained or to one of the remaining classes. Each CNN is trained on single depth maps belonging to the considered class and depth maps sampled from pool of images from remaining classes. As the number of single depth maps in a typical dataset for depth-based action recognition is considerable, it is possible to learn CNNs without overfitting and with good generalization capabilities. CNNs trained in such a way are then used to extract features representing person shapes in depth maps. In our approach the features are extracted using the outputs from the penultimate layer of the convolutional neural network. The number of the features is equal to one hundred. Since the number of frames in depth maps sequences representing actions is not identical, the lengths of multivariate time-series are not the same. The discussed CNN-based features can be used alone or together with handcrafted features depending on the configuration of the algorithm.

In this work the input depth maps were scaled to sizes 64×64 pixels. The input of convolutional neural network is a $4 \times 64 \times 64$ tensor consisting of two consecutive depth maps, and orthogonal projection of the input depth map onto xz and yz planes. The convolutional layer C1 consists of sixteen 5×5 convolutional filters that are followed by a subsampling layer, see Fig. 1. Next convolutional layer C2 operates on sixteen feature maps of size 28×28. It comprises sixteen 5×5 convolutional filters that are followed by a subsampling layer. Its output is sixteen feature maps of size 12×12. Subsequent fully connected layer FC has one hundred neurons. At the learning stage, the output of the CNN is a softmax layer with number of neurons equal to the number of actions to be recognized. Such a network has been learned on depth maps from the training subsets. After the training, the layer before the softmax has been used to extract features representing person's shape. An action consisting of a number of depth maps is described by a multivariate time-series of length equal the number of frames and dimension equal to one hundred.

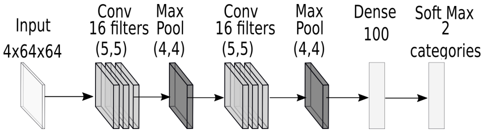


Figure 1: Flowchart of the CNN for learning frame-features.

2.1.2 Handcrafted Frame-Features

Aside from the CNN-based features, for each frame we calculate handcrafted features. At the beginning the depth maps are projected on y and z axes [3]. This means that we obtain side-view and top projections of the depth maps, i.e. projected depth maps. Only non-zero depth map values, i.e. pixels representing the extracted person are utilized to calculate the features. The following features were calculated on such depth maps: area ratio (calculated only for frontal depth map in axes x, y, expressing the area occupied by the person to total number of pixels in the depth map), standard deviation (axes x, y, z), skewness (axes x, y, z), correlation (xy, xz and zy axes), $x-$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 number of handcrafted features describing a single depth map is equal to twelve. An action represented by a number of depth maps is described by a multivariate time-series of length equal the number of frames and dimension equal to twelve.

2.2 CNN-based Time-series Classification and Representation

2.2.1 Multi-channel, temporal CNN for Time-series Classification and Feature Extraction.

In multi-channel, temporal CNNs the 1D convolutions are applied in the temporal domain. In this work, the time-series (TS) of the handcrafted frame-features were used to train a multi-channel CNN. The number of channels is equal to the number of the handcrafted frame-features, i.e. to twelve. The multivariate time-series were interpolated to the length equal to 128. This means that regardless of the length of the multivariate time-series, the length of time-series representing any action is equal to 128. Cubic-spline algorithm has been used to interpolate the TS to the common length. The first layer of the MC CNN is a filter (feature detector) operating in time domain. Having on regard that the amount of the training data in current datasets for depth-based action recognition is quite small, the neural network consists of two convolutional layers, each with 8×1 filter and 4×1 max pool, see Fig. 2. The number of neurons in the dense layer is equal to one hundred. The neural network has been trained on time-series of all training data sequences. The number of output neurons is equal to number of the classes. Nesterov Accelerated Gradient (NesterovMomentum) has been used to train the network, in 1500 iterations, with momentum set to 0.9, dropout equal to 0.5, learning rate equal to 0.001, L1 parameter set to 0.001

and dropout set to 0.5. The network trained in such a way was used to classify the actions as well as to extract the features. After training, the output of the dense layer has been used to extract the features.

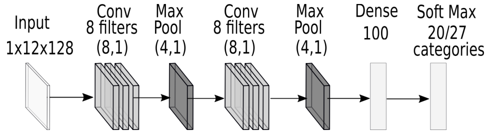


Figure 2: Flowchart of the multi-channel CNN for multivariate time-series modeling and classification.

2.2.2 Time-series Augmentation.

The multi-channel CNN has been trained on augmented time-series. Additional four time-series were generated for each input sequence through extracting: (i) the first sixteen data and then interpolating to 8, (ii) the first sixteen data and then interpolating to 32, (iii) the last sixteen data and then interpolating to 8, (iv) the last sixteen data and then interpolating to 32, and then adding such interpolated subsequences to the input sequence, and finally interpolating the concatenated TS to the length equal to 128. Afterwards, two additional time-series were generated by scaling the sequences in time domain by 2 and 0.5.

2.3 Statistical Features of Time-series

2.3.1 Statistical Temporal Features.

For each multivariate time-series of handcrafted features, as well as each multivariate time-series of CNN-based frame-features, representing the action with different number of frames 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 extended statistical temporal features (ESTF). The motivation of using skewness was to include a parameter describing asymmetry in random variable's probability distribution with respect to normal distribution. The ESTF has size $4 \times 12 = 48$ for time-series of hand-crafted features and size $4 \times 100 = 400$ for time-series of CNN-based frame-features.

2.3.2 Pointwise Nonlinearity.

Except of features described above, we calculated pointwise nonlinearity. The discussed feature is calculated only for handcrafted frame-features. At the beginning a TS is interpolated to the length equal to 128. After FFT-based suppressing high level frequencies, it is scaled such that $\max(t_s) - \min(t_s) = 1$, where t_s denotes 128-element time-series. Such scaled time-series t_{ss} is convolved with filter $f = [-0.25\ 0.25\ 1.0\ 0.2\ 0.25]$: $r_s = t_{ss} * f$, where $*$ denotes the convolution. The pointwise feature is calculated by dividing pointwise r_s by t_{ss}. Finally, we calculate the mean, the median and the max value. This way the pointwise nonlinear features of time-series with handcrafted frame-features have sizes equal to 36.

2.4 Multi-class Classifiers to Construct Ensemble

The features described in Subsections 2.2 and 2.3 were used to train multi-class classifiers with one-hot encodings, see Fig. 3 that depicts a flowchart of single ensemble classifier. Having on regard that for each action an action-specific classifier to extract depth map features has been trained, the number of such classifiers is equal to the number of actions to be recognized. The CNNs operating on depth maps (Subsect. 2.1) deliver time-series of CNN-based frame features, on which we determine extended statistical temporal features (Subsect. 2.3). Extended statistical temporal features, pointwise nonlinearity features (Subsect. 2.3) and multi-channel CNN-based features (Subsect. 2.2) are concatenated together, and then concatenated with the mentioned above statistical features of temporal CNN-based features. The features were selected using recursive feature selection (RFE) algorithm. The multiclass classifiers with one hot encoded outputs are finally used in an ensemble responsible for classification of actions.

2.5 Ensemble of Classifiers

Figure 4 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. 3. As we can see, our class-specific ESTF are concatenated with CFV (common feature vector), and then used to train multi-class classifiers.

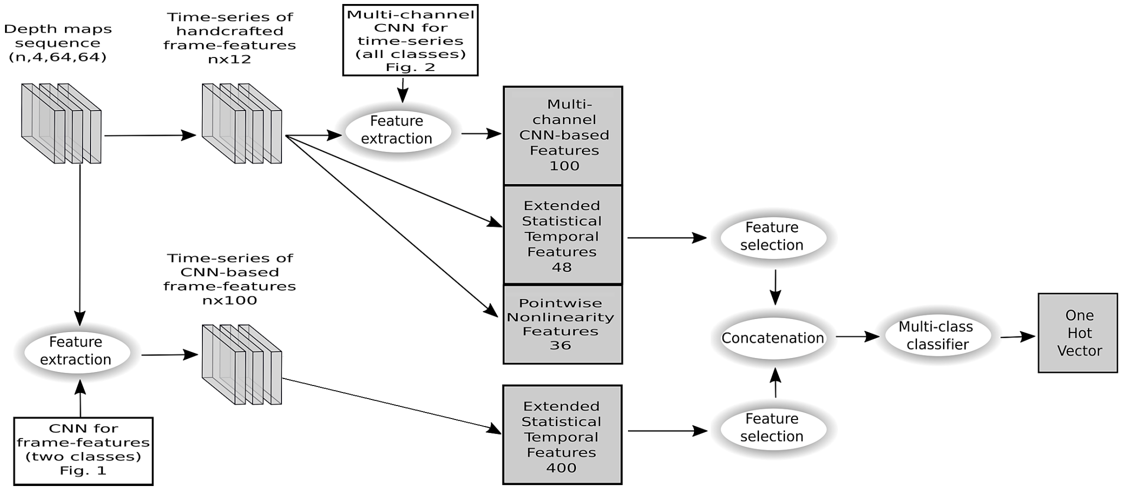


Figure 3: Multi-class classifier to construct ensemble.

3. EXPERIMENTAL RESULTS AND DISCUSSION

The proposed algorithms have 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 has been employed to get the test subset. In the discussed classification setting, half of the subjects are used for the training, and the rest for the testing, which is different from evaluation protocols based on AS1, AS2 and AS3 data splits and averaging the classification accuracies over such data splits. 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 to bigger variations across the same actions performed by different subjects. The cross-subject evaluation protocol [5, 8] has been applied in all evaluations. This procedure is different from the scheme utilized in [9], in which more subjects were in the training subset.

The UTD-MHAD dataset comprises 27 different actions performed by eight subjects (four females and four males).

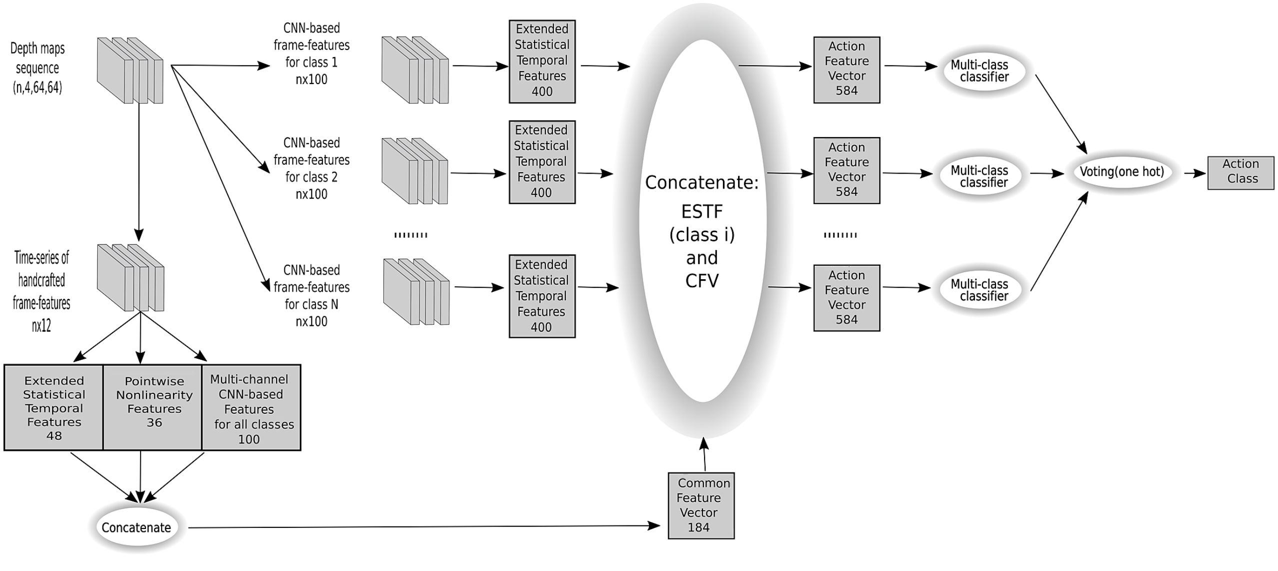


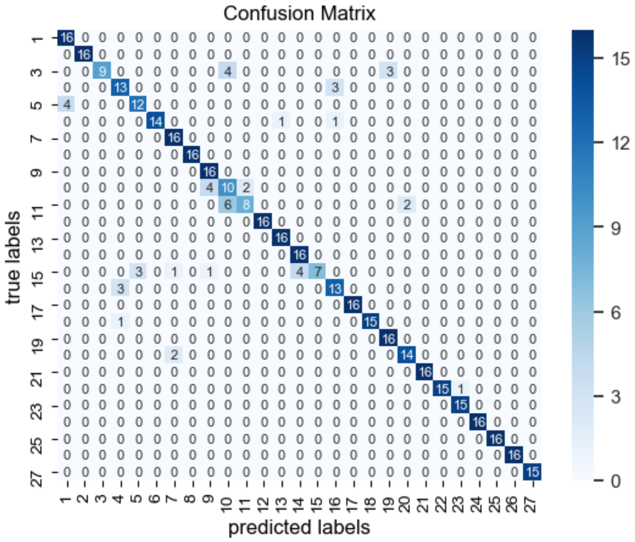
Figure 4: Ensemble operating on handcrafted features concatenated with class-specific features.

Each 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.

Table 1 presents experimental results that were achieved on the UTD-MHAD dataset. Results achieved by the multi-class CNN classifier (Sect. 2.2.1) are presented in first two rows. In next two rows, results achieved by a basic classifier, see Fig. 3, operating on multi-channel CNN features and extended statistical temporal features (ESTF) that are calculated on handcrafted frame-features (Subs. 2.1.2) are shown. In the subsequent rows we can see results achieved by the ensemble, c.f. single ensemble classifier on Fig. 3. The pointwise nonlinear features are denoted as nonlin. As we can notice, the best results were achieved by ensemble operating on conv + MC CNN + ESTF + nonlin. features. Figure 5 depicts the confusion matrix, which has been obtained by the ensemble operating on conv + MC CNN + ESTF + nonlin. features.

Table 1: Recognition performance on UTD-MHAD dataset.

f. set	TS aug.	deep ens.	f. sel.	f. num.	Accuracy	Precision	Recall	F1-score
MC CNN	no	-	-	-	0.8023	0.8200	0.8023	0.7978
MC CNN	yes	-	-	-	0.8233	0.8317	0.8233	0.8191
conv+ESTF	-	yes	-	178	0.7884	0.8060	0.7884	0.7859
conv+nonlin	-	yes	RFE	166	0.7837	0.7945	0.7837	0.7741
conv+ESTF+nonlin	-	yes	RFE	214	0.8372	0.8477	0.8372	0.8336
conv+MC CNN	no	yes	RFE	230	0.8279	0.8409	0.8279	0.8233
conv+MC CNN	yes	yes	RFE	230	0.8302	0.8505	0.8302	0.8261
conv+MC CNN+nonlin	no	yes	RFE	266	0.8512	0.8618	0.8512	0.8481
conv+MC CNN+nonlin	yes	yes	RFE	230	0.8651	0.8727	0.8651	0.8577
conv+MC CNN+ESTF	no	yes	RFE	230	0.8605	0.8673	0.8605	0.8578
conv+MC CNN+ESTF	yes	yes	RFE	230	0.8581	0.8693	0.8581	0.8578
conv+MC CNN+ESTF+nonlin	no	yes	RFE	230	0.8674	0.8786	0.8674	0.8654
conv+MC CNN+ESTF+nonlin	yes	yes	RFE	230	**0.8930**	**0.9045**	**0.8930**	**0.8885**



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
1	16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	9	0	0	0	0	0	0	4	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0
	0	0	0	13	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0
5	4	0	0	0	12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	14	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	4	10	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	6	8	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	16	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	16	0	0	0	0	0	0	0	0	0	0	0	0	0
15	0	0	0	0	3	0	1	0	1	0	0	0	0	4	7	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	13	0	0	0	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	16	0	0	0	0	0	0	0	0	0	0
	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	15	0	0	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	16	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	14	0	0	0	0	0	0	0
21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	16	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	15	1	0	0	0	0
23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	15	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	16	0	0	0
25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	16	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	16	0
27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	15

Figure 5: Confusion matrix on UTD-MHAD, obtained by ensemble operating on conv+MC CNN+ESTF+nonlin. features.

Table 2 presents experimental results that were achieved on the MSR Action 3D dataset. As we can observe, the best results were achieved by ensemble operating on conv + MC CNN + ESTF + nonlin. features. Comparing the results achieved using conv + MC CNN + ESTF + nonlin. features and conv ESTF + nonlin. features, i.e. by the use of MC CNN features and no use of MC CNN features we can observe a considerable improvement of the classification performance.

In general, the features extracted by the MC CNN give quite good results. Deep ensemble gives better results in comparison to the basic classifier of actions. Time-series augmentation leads to better results. RFE-based feature selection permits achieving better results.

Table 2: Recognition performance on MSR Action 3D dataset.

f. set	TS aug.	deep ens.	f. sel.	f. num.	Accuracy	Precision	Recall	F1-score
MC CNN	no	-	-	-	0.8069	0.7964	0.8069	0.7877
MC CNN	yes	-	-	-	0.8505	0.8637	0.8509	0.8426
conv+ESTF	-	yes	-	178	0.8909	0.9028	0.8909	0.8845
conv+nonlin	-	yes	RFE	166	0.8909	0.9032	0.8909	0.8814
conv+ESTF+nonlin	-	yes	RFE	214	0.8945	0.9121	0.8945	0.8845
conv+MC CNN	no	yes	RFE	230	0.8873	0.9074	0.8873	0.8763
conv+MC CNN	yes	yes	RFE	230	0.9055	0.9176	0.9055	0.9059
conv+MC CNN+nonlin	no	yes	RFE	266	0.8800	0.8989	0.8800	0.8704
conv+MC CNN+nonlin	yes	yes	RFE	230	0.9091	0.9176	0.9091	0.9074
conv+MC CNN+ESTF	no	yes	RFE	230	0.8945	0.9076	0.8945	0.8859
conv+MC CNN+ESTF	yes	yes	RFE	230	0.9091	0.9192	0.9091	0.9088
conv+MC CNN+ESTF+nonlin	no	yes	RFE	230	0.9018	0.9183	0.9018	0.8959
conv+MC CNN+ESTF+nonlin	yes	yes	RFE	230	**0.9200**	**0.9288**	**0.9200**	**0.9191**

Table 3 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 skeleton data. Methods based on skeleton modality usually achieve better results in comparison to methods relying on depth data only. Despite the fact that our method is based on depth modality, we evoked the recent skeleton-based methods to show that it outperforms many of them. The methods based on depth data have wide range applications since not all sensors or cameras delivering depth modality have support for skeleton extraction. Our method is considerably better than the WHDMM+3DConvNets method that employs weighted hierarchical depth motion maps (WHDMMs) and three 3D ConvNets. The WHDMMs are employed at several temporal scales to encode spatiotemporal motion patterns of actions into 2D spatial structures. In order to provide sufficient amount of training data, the 3D points are rotated and then used to synthesize new exemplars. In contrast, our algorithm operates on CNN features that are concatenated with handcrafted features. The improved performance of our method may suggest that the proposed method has better viewpoint tolerance in comparison to depth-based algorithms, including [8].

Table 3: Comparative recognition performance of the proposed method with recent algorithms on MHAD dataset.

Method	Modality	Accuracy [%]
JTM [10]	skeleton	85.81
SOS [11]	skeleton	86.97
Kinect & inertial [3]	skeleton	79.10
Struct. body [12]	skeleton	66.05
Struct. part [12]	skeleton	78.70
Struct. joint [12]	skeleton	86.81
Struct. SzDDI [12]	skeleton	89.04
WHDMMs+ConvNets [8] [12]	depth	73.95
Proposed Method	depth	**89.30**

Table 4 illustrates the classification performance of the proposed method in comparison to previous depth-based methods on the MSR-Action3D dataset. The classification performance of the proposed framework has been determined using the cross-subject evaluation [13], where subjects 1, 3, 5, 7, and 9 were employed for training and subjects 2, 4, 6, 8, and 10 were utilized for testing. As we can notice, the proposed method achieves better classification accuracy than recently proposed method [10], and it has worse performance in comparison to recently proposed methods [11] (Split II) and [12, 14]. One of the main reasons for this is limited amount of training samples in the MSR-Action3D dataset. To cope with such a limitation, Wang et al. generated synthesized training samples on the basis of 3D points. This means that the discussed algorithm is not based on depth maps only. Comparing the results from Tab. 3 and Tab. 4 we can notice that results achieved by the WHDMM algorithm on UTD-MHAD dataset are worse than results achieved by the proposed algorithm.

Table 4: Comparative recognition performance of the proposed method with recent algorithms on MSR Action 3D dataset.

Method	Split	Modality	Acc. [%]
3DCNN [10]	Split II	depth	84.07
DMMs [4]	Split II	depth	88.73
PRNN [11]	Split II	depth	94.90
WHDMM+CNN [14]	Split I	depth	100.00
S DDI [12]	Split I	depth	100.00
Proposed Method	Split I	depth	92.00

4. CONCLUSIONS

In this paper we presented novel algorithms for human action recognition on depth maps. The novelty comprises the learning multi-channel CNN on small amount of dataset using time-series augmentation. We compared the classification performances of algorithms using multi-channel CNN features and not using such features and demonstrated experimentally that multi-channel CNN allows us to obtain considerable improvement of the classification performance. The presented algorithm achieves promising results. We demonstrated experimentally that our algorithm outperforms several recent skeleton-based methods. The source code is available at `https://github.com/tjacek/DeepActionLearning`.

ACKNOWLEDGMENTS

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

REFERENCES

[1] B. Liang and L. Zheng, “A survey on human action recognition using depth sensors,” in *Int. Conf. on Digital Image Computing: Techniques and Appl.*, pp. 1–8, 2015.

[2] W. Li, Z. Zhang, and Z. Liu, “Action recognition based on a bag of 3D points,” in *IEEE Int. Conf. on Computer Vision and Pattern Rec. - Workshops*, pp. 9–14, 2010.

[3] C. Chen, R. Jafari, and N. Kehtarnavaz, “UTD-MHAD: A multimodal dataset for human action recognition utilizing a depth camera and a wearable inertial sensor,” in *IEEE Int. Conf. on Image Processing*, pp. 168–172, Sept 2015.

[4] X. Yang, C. Zhang, and Y. L. Tian, “Recognizing actions using depth motion maps-based histograms of oriented gradients,” in *Proc. of the 20th ACM Int. Conf. on Multimedia*, pp. 1057–1060, ACM, 2012.

[5] L. Xia and J. Aggarwal, “Spatio-temporal depth cuboid similarity feature for activity recognition using depth camera,” in *IEEE Int. Conf. on Computer Vision and Pattern Recognition*, pp. 2834–2841, 2013.

[6] Y. Zheng, Q. Liu, E. Chen, Y. Ge, and J. L. Zhao, “Time series classification using multi-channels deep convolutional neural networks,” in *Web-Age Information Management*, pp. 298–310, Springer Int. Publ., (Cham), 2014.

[7] Y. Zheng, Q. Liu, E. Chen, Y. Ge, and J. L. Zhao, “Exploiting multi-channels deep convolutional neural networks for multivariate time series classification,” *Frontiers of Computer Science* **10**(1), pp. 96–112, 2016.

[8] P. Wang, W. Li, Z. Gao, J. Zhang, C. Tang, and P. Ogunbona, “Action recognition from depth maps using deep convolutional neural networks,” *IEEE Trans. on Human-Machine Systems* **46**(4), pp. 498–509, 2016.

[9] L. Xia, C.-C. Chen, and J. Aggarwal, “View invariant human action recognition using histograms of 3D joints.,” in *CVPR Workshops*, pp. 20–27, 2012.

[10] P. Wang, W. Li, C. Li, and Y. Hou, “Action recognition based on joint trajectory maps with convolutional neural networks,” *Knowledge-Based Syst.* **158**, pp. 43 – 53, 2018.

[11] Y. Hou, Z. Li, P. Wang, and W. Li, “Skeleton optical spectra-based action recognition using convolutional neural networks,” *IEEE Trans. on Circuits and Systems for Video Technology* **28**(3), pp. 807–811, 2018.

[12] P. Wang, S. Wang, Z. Gao, Y. Hou, and W. Li, “Structured images for RGB-D action recognition,” in *IEEE Int. Conf. on Computer Vision Workshops (ICCVW)*, pp. 1005–1014, 2017.

[13] Y. Wu, “Mining actionlet ensemble for action recognition with depth cameras,” in *IEEE Int. Conf. on Computer Vision and Pattern Recognition*, pp. 1290–1297, 2012.

[14] P. Wang, W. Li, Z. Gao, J. Zhang, C. Tang, and P. O. Ogunbona, “Action recognition from depth maps using deep convolutional neural networks,” *IEEE Trans. on Human-Machine Systems* **46**(4), pp. 498–509, 2016.