Embedded Features for 1D CNN-based Action Recognition on Depth Maps

Jacek Trelinski and Bogdan Kwolek
AGH University of Science and Technology, 30 Mickiewicza, 30-059 Krakow, Poland

Keywords: Action Recognition on Depth Maps, Convolutional Neural Networks, Feature Embedding.

Abstract: In this paper we present an algorithm for human action recognition using only depth maps. A convolutional autoencoder and Siamese neural network are trained to learn embedded features, encapsulating the content of single depth maps. Afterwards, statistical features and multichannel 1D CNN features are extracted on multivariate time-series of such embedded features to represent actions on depth map sequences. The action recognition is achieved by voting in an ensemble of one-vs-all weak classifiers. We demonstrate experimentally that the proposed algorithm achieves competitive results on UTD-MHAD dataset and outperforms by a large margin the best algorithms on 3D Human-Object Interaction Set (SYSU 3DHOI).

1 INTRODUCTION

Human action recognition (HAR) can be defined as capability of determining whether a given action occurred in image or depth sequence. It has been intensively studied in the last decade, especially because this is a challenging problem, but also due to possible applications that can benefit from it. However, due to several difficulties, including non-rigid shape of the humans, intra-class variations, viewpoint variations, occlusions and plenty another challenges and environmental complexities, the current algorithms have poor performance in comparison to human ability to recognize and to understand human motions and actions (Liang and Zheng, 2015; Wang et al., 2020).

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. Having on regard that Kinect motion sensor is capable of extracting the depth maps in poor illumination conditions or even in darkness, considerable attention is devoted to algorithms operating on depth maps. When action performers are captured using 2D cameras then one dimension is lost during the acquisition, which causes the loss of important information. For the same reason, 3D-based approaches provide higher accuracy than 2D-based approaches. Most of the current approaches to action recognition on depth maps are based on the skeleton (Ren et al., 2020). The number of approaches, relying on depth maps only, particularly deep learning-based is very limited (Wang et al., 2020). Despite that skeleton-based methods usually achieve better results in comparison to algorithms using only depth maps, they can fail in many scenarios due to skeleton extraction failure.

In order to stimulate the research as well as to facilitate development and evaluation of new algorithms, several RGB-D benchmark datasets have been acquired in the last decades (Li et al., 2010; Chen et al., 2015; Hu et al., 2015). Currently available datasets for 3D action recognition usually have 10, 20, 27 or a little more categories of actions, which were performed by a dozen or several dozen performers, and each action has been repeated a few times. The MSR Action 3D dataset (Li et al., 2010), is one of the most frequently used benchmarks in the research as well as in evaluation of the algorithms. The recently introduced UTD-MHAD dataset (Chen et al., 2015) 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. In SYSU 3D Human-Object Interaction (3DHOI) dataset (Hu et al., 2015) each action involves a kind of human-object interactions. Deep learning-based methods 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 necessitates appropriate equipment and computational resources. Due to limited number of depth sequences in currently available RGB-D datasets, which is typically smaller than one thousand, the recogni-

Trelinski, J. and Kwolek, B.
Embedded Features for 1D CNN-based Action Recognition on Depth Maps.
DOI: 10.5220/0010340105360543
In *Proceedings of the 16th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2021) - Volume 4: VISAPP*, pages 536-543
ISBN: 978-989-758-488-6

tion of actions on the basis of 3D depth maps is very challenging.

Traditional approaches to activity recognition on depth maps rely on the handcrafted feature-based representations (Yang et al., 2012; Xia and Aggarwal, 2013). In contrast to handcrafted representation-based approaches, in which actions are represented by engineered features, learning-based algorithms are capable of discovering the most informative features automatically from the raw data. Such deep learning-based methods permit processing images/videos in their raw forms and thus they are capable of automating the process of feature extraction and classification. These methods employ trainable feature extractors and computational models with multiple layers for action representation and recognition.

In this work we demonstrate experimentally that despite the limited amount of training data, i.e. action sequences in currently available datasets, it is possible to learn features with highly discriminative power. A convolutional autoencoder and Siamese neural network have been trained to extract features on single depth maps. Afterwards, statistical features and multi-channel 1D CNN features were extracted on multivariate time-series of features to represent actions on depth map sequences. As far as we know, the multi-channel 1D convolutional neural networks (Zheng et al., 2014; Zheng et al., 2016) have not been utlized in human action recognition. The recognition of the actions is done by a voting-based ensemble operating on one-hot encodings of one-vs-all weak classifiers.

2 THE ALGORITHM

A characteristic feature of the proposed approach is that it does not require skeleton. Thanks to using depth maps only, our algorithm can be employed on depth data provided by stereo cameras, which can deliver the depth data for persons being at larger distances to the sensors. It is well known that the Kinect sensor fails to estimate the skeleton in several scenarios. In the next Section, we demonstrate experimentally that despite no use of the skeleton, our algorithm achieves better accuracies than several skeleton-based algorithms. In the proposed approach, various features are learned in different domains, like single depth map, time-series of embedded features, and final decision is taken on the basis of voting of one-vs-all weak classifiers.

The algorithm has been evaluated on MSR Action 3D, UTD-MHAD and SYSU 3DHOI datasets. Since in SYSU 3DHOI dataset the performers are not extracted from depth maps, we extracted the subjects. For each depth map we determined a window surrounding the person, which has then been scaled to the required input shape.

In Subsection 2.1 we present features describing the person's shape in single depth maps. Afterwards, in Subsection 2.2 we present features representing multivariate time-series. In Subsection 2.3 we discuss multi-class classifiers to construct ensemble. Finally, in Subsection 2.4 we describe the ensemble.

2.1 Embedded 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 CNNs operating on single depth maps or pairs of depth maps to extract informative frame-features. Because the number of frames in the current benchmark datasets for RGBD-based action recognition is pretty large, deep feature representations can be learned. Given an input depth map sequence $x = \{x_1, x_2, \ldots, x_T\}$, we encode each depth map x_i using a CNN backbone f into a feature $f(x_i)$, which results in a sequence of embedded feature vectors $f(x) = \{f(x_1), f(x_2), \ldots, f(x_T)\}$. The dimension of such embedding for a depth map sequence is $T \times D_f$, where D_f is size of the embedded vector. In the next subsection we detail how frame-features were embedded without supervision using a convolutional autoencoder. Afterwards, we explain how frame-features were calculated using a Siamese neural network.

2.1.1 Unsupervised Extraction of Frame-features using Convolutional Autoencoder

An autoencoder is a type of artificial neural network that projects a high-dimensional input into a latent low-dimensional code (encoder), and then carries out a reconstruction of the input using such a latent code (the decoder) (Hinton and Salakhutdinov, 2006). To achieve this the autoencoder learns a hidden representation for a set of input data, typically through compression (dimensionality reduction), by learning to ignore less informative information. This means that the autoencoder tries to generate from such a reduced encoding an output representation that is close as possible to its input. When the hidden representation uses fewer dimensions than the input, the encoder carries out dimensionality reduction. An autoencoder consists of an internal (hidden) layer that stores a compressed representation of the input, as well as an encoder that maps the input into the code, and a decoder that maps the code to a reconstruction of the original input. The encoder compresses the input and produces the code, whereas the decoder reconstructs the input using only

this code. Learning to replicate its input at its output is achieved by learning a reduction side and a reconstructing side. Autoencoders are considered as unsupervised learning technique since no explicit labels are needed to train them. Once such a representation with reduced dimensionality is learned, it can then be taken as input to a supervised algorithm that can then be trained on the basis of a smaller labeled data subset.

We extracted frame-features using encoder/decoder paradigm proposed in (Masci et al., 2011). We implemented a convolutional autoencoder (CAE) in which the input depth map is first transformed into a lower dimensional representation through successive convolution operations and rectified linear unit (ReLU) activations and afterwards expanded back to its original size using deconvolution operations. The mean squared error, which measures how close the reconstructed input is to the original input has been used as the loss function in the unsupervised learning. The network has been trained using Adam optimizer with learning rate set to 0.001. After training, the decoding layers of the network were excluded from the convolutional autoencoder. The network trained in such a way has been used to extract low dimensional frame-features. The depth maps acquired by the sensor were projected two 2D orthogonal Cartesian planes to represent top and side view of the maps. On training subsets we trained a single CAE for all classes. The convolutional autoencoder has been trained on depth maps of size $3 \times 64 \times 64$.

The CAE network architecture is shown in Fig. 1. The network consists of two encoding layers and two associated decoding layers. The size of depth map embedding is equal to 100.

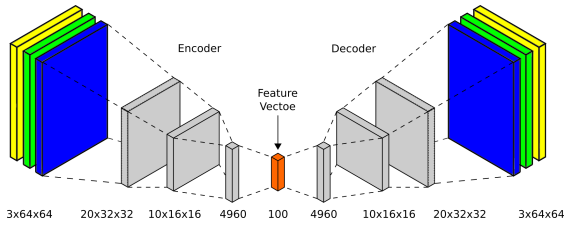

Figure 1: Architecture of convolutional autoencoder.

2.1.2 Semi-supervised Learning of Frame Similarity

Siamese Neural Networks (SNNs) are a class of neural networks that contain multiple instances of the same model and share same architecture and weights (Chopra et al., 2005). In contrast to MLPs that utilize loss functions calculating errors between outputs and target values, the SNNs use objectives that compare the feature vectors of pairs of the exemplars. They have been used for dimensionality reduction (Hadsell et al., 2006) as well for one-shot image classification (Koch et al., 2015), i.e. recognition on the basis of one training example per class. They are trained with a collection of sample pairs with the same or different categories. Sharing weights across the sub-networks results in smaller number of learned parameters. Thus, this architecture shows its strength when it has to learn on the basis of limited data.

In the proposed approach a separate Siamese neural network is trained for each action class to extract features distinguishing a given class from all remaining classes, same as in one-vs-all multi-class classification. In other words, each Siamese neural network is trained to decide whether the considered depth map belongs to the class for which the network had been trained or to one of the remaining classes.

The central idea behind SNNs is to learn an embedding, where similar image pairs are close to each other and dissimilar image pairs are separated by a distance that depends on a parameter called margin. Let us assume that our aim is to learn a SNN on a training set $\{x_i, y_i\}_{i=1}^{T}$, where $x_i \in \mathbb{R}^n$ and y_i are class labels. The Siamese network produces a feature embedding $f(x, \theta_f)$ that is defined as $f : \mathbb{R}^n \times \mathbb{R}^k \rightarrow \mathbb{R}^m$, where $\theta_f \in \mathbb{R}^k$ stands for parameters of the network. The aim of the learning is to seek for the parameter vector θ_f such that the embedding produced through f has desirable properties, and particularly it places similar examples nearby. The SNN can extract information from the available data and determine such an embedding without requiring specific information about the categories. Thus, learning the Siamese network is achieved in a weakly-supervised scheme using only pairs of data instances labeled as matching or non-matching.

Let us consider a pair of images (I_p, I_q) that contains person performing the same action, and a pair of images (I_p, I_r), which belong to different action categories. The Siamese neural network maps such image pairs into embedded space (x_p, x_q, x_r) such x_p and x_q are close, whereas x_p and x_r are further apart. Such an embedding can be achieved on the basis of a contrastive loss function L, which expresses how well the function f is capable of placing similar image representations in the close proximity and keep dissimilar image representations distant. The contrastive loss function $L(\theta)$ can be expressed as follows:

$$L(\theta) = \underbrace{\sum_{(x_p, x_q)} L_q(x_p, x_q)}_{\substack{\text{penalty term} \\ \text{for similar images}}} + \underbrace{\sum_{(x_p, x_r)} L_r(x_p, x_r)}_{\substack{\text{penalty term} \\ \text{for dissimilar images}}} \quad (1)$$

where

$$L_q(x_p,x_q) = \left\|x_p - x_q\right\|_2^2$$
$$L_r(x_p,x_r) = \max\left(0, m - \left\|x_p - x_r\right\|_2\right)^2 \quad (2)$$

The penalty term L_q penalizes the pair (x_p,x_q) that is too apart, whereas L_r penalizes the pair (x_p,x_r) that is closer than a margin m. Thus, dissimilar image pairs contribute to the loss function if their distances are within the margin m. Given the label y indicating whether the images are similar or dissimilar the margin-based loss function can be expressed as follows:

$$L(\theta,x_i,x_j) = y\left\|f(x_i) - f(x_j)\right\|_2^2 + (1-y)\max(0, m - \left\|f(x_i) - f(x_j)\right\|_2)^2 \quad (3)$$

where function f performs a forward propagation through the Siamese sub-network. The loss function penalizes positive pairs by the squared Euclidean distances and negative pairs by the squared difference between the margin m and Euclidean distance for pairs having distance less than the margin m. Figure 2 contains diagram of the utilized Siamese neural network. The network has been trained using Adam optimizer with learning rate set to 0.00006.

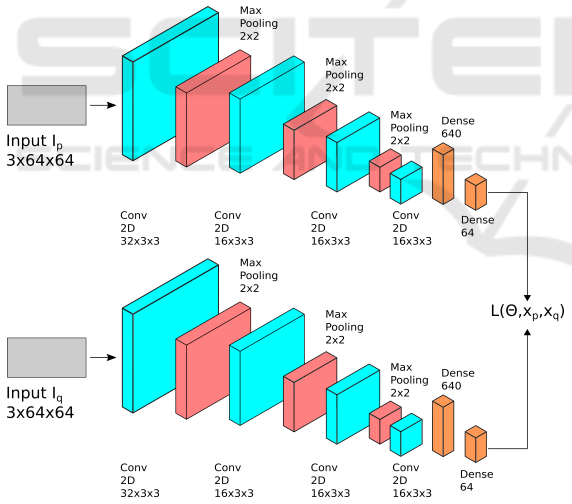

Figure 2: Architecture of Siamese neural network.

2.2 Features of Time-series

On the basis of depth map sequences representing human actions the neural networks that were discussed above produce multivariate time-series. Having on regard that depth map sequences differ in length, such variable length time-series were interpolated to a common length. In the next two Subsections we explain how features for multivariate time-series are determined.

2.2.1 Statistical Features of Time-series

For each multivariate time-series of features, which were extracted by the convolutional autoencoder, 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 motivation of using skewness was to include a parameter describing asymmetry in random variable's probability distribution with respect to normal distribution. The multivariate time-series are of size $4 \times 100 = 400$.

2.2.2 Multi-channel, Temporal 1D CNN

In multi-channel, temporal CNNs (MC CNNs) the 1D convolutions are applied in the temporal domain. In this work, the time-series (TS) of frame-features that were extracted by the Siamese neural network have been used to train multi-channels 1D CNNs. The number of channels is equal to 64, see Fig. 2. The multivariate time-series were interpolated to the length equal to 100. Cubic-spline algorithm has been utilized to interpolate the TS to such a 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, 4×1 and 2×1 max pools, respectively, see Fig. 3. The number of neurons in the dense layer is equal to 100. For each Siamese time-series of 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 the training, the output of the dense layer has been used to extract the features.

2.3 Multi-class Classifiers to Construct Ensemble

The features described in Subsections 2.2.1 and 2.2.2 were used to train multi-class classifiers with softmax encoding, see Fig. 4. Having on regard that for each class 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 convolutional Autoencoder (Subsect. 2.1.1) operating on sequences of depth maps delivers time-series

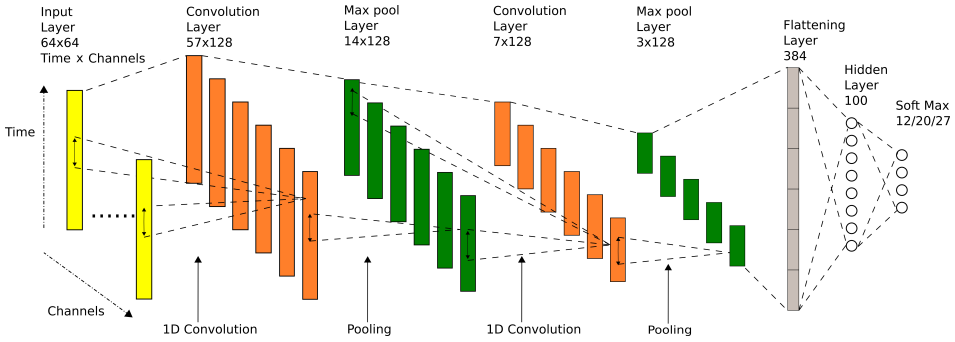

Figure 3: Flowchart of the multi-channel CNN for multivariate time-series modeling.

of AE-based frame-features, on which we determine statistical temporal features (Subsect. 2.2.1). The base networks of the Siamese (Subsect. 2.1.2) operating on sequences of depth maps deliver time-series of features (one feature vector per frame), which are further processed by the multi-channel, temporal neural network (Subsect. 2.2.2). It delivers feature vectors of size 100, see Fig. 4, which are then concatenated with statistical features of size 400. The multi-class classifiers delivering at the outputs the softmax-encoded class probability distributions are finally used in an ensemble responsible for classification of actions.

2.4 Ensemble of Classifiers

Figure 4 depicts the ensemble for action classification. The final decision is calculated on the basis of voting of the classifiers. As we can see, the statistical features that are common for all actions are concatenated with class-specific features, and then used to train multi-class classifiers.

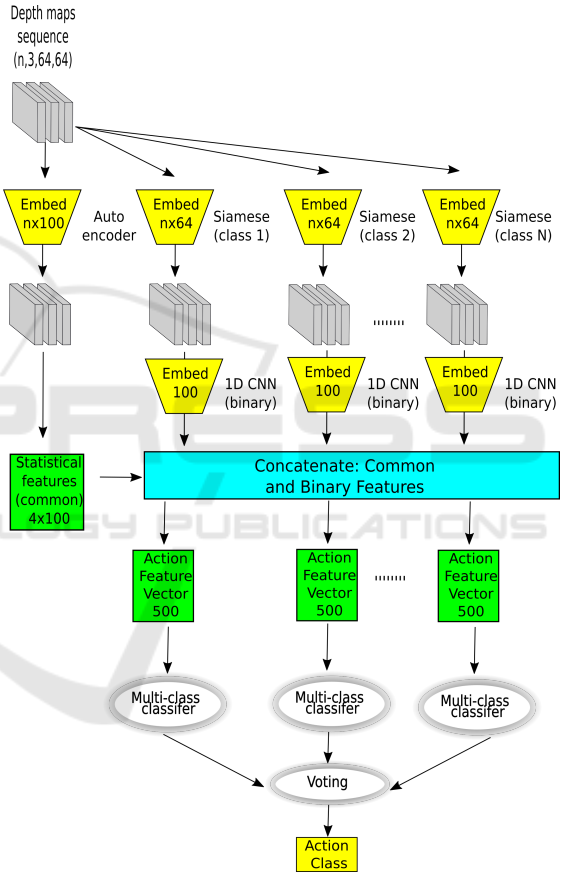

Figure 4: Ensemble operating on features extracted by convolutional autoencoder, concatenated with class-specific features that were extracted by Siamese neural networks.

3 EXPERIMENTAL RESULTS

The proposed algorithm has been evaluated on three publicly available benchmark datasets: MSR Action 3D dataset (Li et al., 2010), UTD-MHAD dataset (Chen et al., 2015) and SYSU 3D Human-Object Interaction Set (SYSU 3DHOI) (Hu et al., 2015). The datasets were selected having on regard their frequent use by action recognition community in the evaluations and algorithm comparisons.

In experiments and evaluations with MSR Action 3D dataset, 557 sequences were investigated. In the evaluation of the algorithm, half of the subjects were utilized for the training, and the rest for the testing, which is dissimilar to evaluation protocols based on AS1, AS2 and AS3 data splits and averaging the classification accuracies over such data splits. It is also worth mentioning that the classification performances achieved in the selected 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 performers. The evaluations were performed according to the cross-subject evaluation protocol (Xia and Aggarwal, 2013; Wang et al., 2016) The discussed evaluation scheme is different from the procedure employed in (Xia et al., 2012), in which more performers were in

the training subset.

The UTD-MHAD dataset contains 27 different actions performed by eight subjects (four females and four males). All actions were performed in an indoor environment with a fixed background. Each performer repeated each action four times. The dataset consists of 861 data sequences and it was acquired using the Kinect sensor and a wearable inertial sensor.

The SYSU 3D Human-Object Interaction (3DHOI) dataset was recorded by the Kinect sensor and comprises 480 RGBD sequences from 12 action categories, including calling with cell phone, playing with a cell phone, pouring, drinking, wearing backpack, packing a backpack, sitting on a chair, moving a chair, taking something from a wallet, taking out a wallet, mopping and sweeping. Actions were performed by 40 subjects. Each action involves a kind of human-object interactions. Some motions actions are quite similar at the beginning since the subjects operate or interact with the same objects, or actions start with the same sub-action, such as standing still. The above mentioned issues make this dataset challenging following the evaluation setting in (Hu et al., 2019), in which depth map sequences with the first 20 subjects were used for training and the rest for testing.

Table 1 presents experimental results that were achieved on the MSR Action 3D dataset. As we can notice, the best results were achieved using logistic regression classifiers and hard voting in the ensemble.

Table 1: Recognition performance on MSR Action 3D dataset (LR - logistic regression, SVM - Support Vector Machine, H - hard voting, S - soft voting).

class.	vot.	Accuracy	Precision	Recall	F1-score
LR	S	0.8836	0.8865	0.8836	0.8716
LR	H	**0.9018**	**0.9076**	**0.9018**	**0.8903**
SVM	S	0.8909	0.8889	0.8909	0.8833
SVM	H	0.8691	0.8704	0.8691	0.8621

Table 2 illustrates the classification performance of the proposed method in comparison to previous depth-based methods on the MSR Action 3D dataset. The classification performance of the proposed framework has been determined using the cross-subject evaluation (Wu, 2012), 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 (Wang et al., 2018), and it has worse performance in comparison to recently proposed method (Bulbul et al., 2019) (Split I). One of the main reasons that our method achieves worse results than (Bulbul et al., 2019) is limited amount of training samples in the MSR Action 3D dataset. In our methods features extracted by CAE and Siamese neural networks are employed, whereas the method mentioned above is based on handcrafted features.

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

Method	Split	Modality	Acc. [%]
3DCNN (Wang et al., 2018)	Split II	depth	84.07
GLAC (Bulbul et al., 2019)	Split I	depth	94.50
Proposed Method	Split I	depth	90.18

Table 3 presents experimental results that were achieved on the UTD-MHAD dataset. As we can observe, the best results were achieved using logistic regression classifiers and hard voting in the ensemble. Figure 5 depicts the confusion matrix.

Table 3: Recognition performance on UTD-MHAD dataset (LR - logistic regression, SVM - Support Vector Machine, H - hard voting, S - soft voting).

class.	vot.	Accuracy	Precision	Recall	F1-score
LR	S	0.8767	0.8912	0.8767	0.8720
LR	H	**0.8791**	**0.8902**	**0.8791**	**0.8760**
SVM	S	0.8744	0.8921	0.8744	0.8724
SVM	H	0.8628	0.8801	0.8628	0.8621

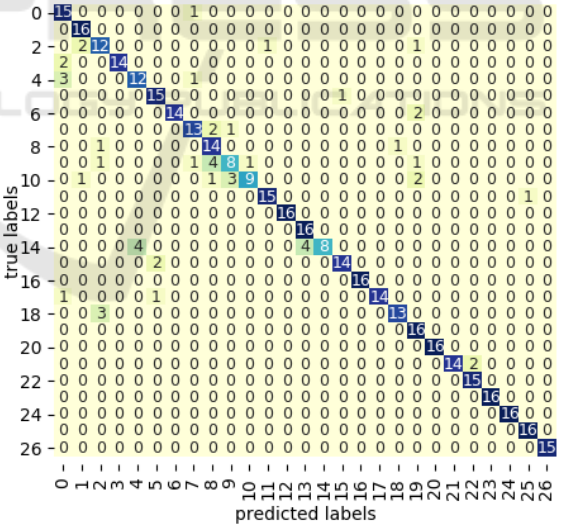

Figure 5: Confusion matrix on UTD-MHAD dataset.

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 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

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

Method	Modality	Accuracy [%]
JTM (Wang et al., 2018)	skeleton	85.81
SOS (Hou et al., 2018)	skeleton	86.97
Kinect & inertial (Chen et al., 2015)	skeleton	79.10
Struct. joint (Wang et al., 2017)	skeleton	86.81
Struct. SzDDI (Wang et al., 2017)	skeleton	89.04
WHDMMs+ConvNets (Wang et al., 2016)(Wang et al., 2017)	depth	73.95
Proposed Method	depth	**87.91**

skeleton-based methods to show that it outperforms many of them. 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 extracts deep features on single frames, and determines features from time-series of such deep features, which 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 (Wang et al., 2016).

Table 5 illustrates results that were achieved on the 3DHOI dataset. As we can observe, the best results were achieved using SVM and hard voting in the ensemble. Figure 6 illustrates the confusion matrix.

Table 5: Recognition performance on SYSU 3DHOI dataset (LR - logistic regression, SVM - Support Vector Machine, H - hard voting, S - soft voting).

class.	vot.	Accuracy	Precision	Recall	F1-score
LR	S	0.8289	0.8294	0.8289	0.8273
LR	H	0.8640	0.8643	0.8640	0.8621
SVM	S	0.9035	0.9077	0.9035	0.9039
SVM	H	**0.9254**	**0.9295**	**0.9254**	**0.9251**

Table 6 presents results achieved by recent algorithms on 3DHOI dataset in comparison to results achieved by our algorithm. As we can observe, our algorithm achieves the best results on this challenging dataset. It is worth noting that method (Wang et al., 2019) relies on depth and skeleton modalities, whereas (Hu et al., 2019) additionally utilizes RGB images jointly with the skeleton data.

We also performed evaluations of a modified algorithm in which instead of statistical features of time series we employed features extracted by a multi channel 1D CNN. As we can notice in Tab. 7 slightly worse results have been achieved by such an algorithm on the 3DHOI dataset. The results on two remaining datasets were also not better.

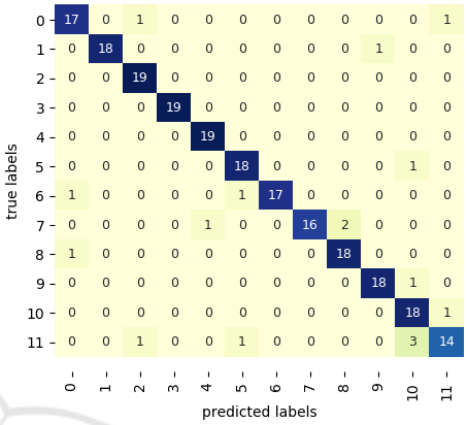

Figure 6: Confusion matrix on 3DHOI dataset.

Table 6: Comparative recognition performance of the proposed method with recent algorithms on 3DHOI dataset.

Method	Modality	Acc. [%]
MSRNN (Hu et al., 2019)	depth+RGB+skel.	79.58
PTS (Wang et al., 2019)	depth+skeleton	87.92
Proposed Method	depth	**92.54**

Table 7: Recognition performance on SYSU 3DHOI dataset achieved by algorithm with 1D CNN-based features instead of statistical features (LR - logistic regression, SVM - Support Vector Machine, H - hard voting, S - soft voting).

class.	vot.	Accuracy	Precision	Recall	F1-score
LR	S	0.8684	0.8687	0.8684	0.8678
LR	H	0.8509	0.8568	0.8509	0.8488
SVM	S	**0.9035**	**0.9091**	**0.9035**	**0.9039**
SVM	H	0.8026	0.8104	0.8026	0.8023

4 CONCLUSIONS

In this paper we presented a novel algorithm for human action recognition on depth maps. The novelty comprises the use of convolutional autoencoder and Siamese network to learn frame-features and then multi-channel 1D CNNs to extract informative features for depth map sequences. In order to cope with depth map sequences with redundant depth maps, in which the human pose are the same in some sub-sequences

for different classes we trained class-specific Siamese neural networks and corresponding multichannel 1D CNNs. The final decision is taken on the basis of the voting. The presented algorithm achieves promising results in comparison to recent algorithms. It achieves considerable gain in recognition accuracy on challenging SYSU 3DHOI dataset. We demonstrated experimentally that our algorithm outperforms several recent skeleton-based methods.

ACKNOWLEDGEMENTS

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

REFERENCES

Bulbul, M., Islam, S., and Ali, H. (2019). 3D human action analysis and recognition through GLAC descriptor on 2D motion and static posture images. *Multim. Tools Appl.*, 78(15).

Chen, C., Jafari, R., and Kehtarnavaz, N. (2015). UTD-MHAD: A multimodal dataset for human action recognition utilizing a depth camera and a wearable inertial sensor. In *IEEE ICIP*, pages 168–172.

Chopra, S., Hadsell, R., and LeCun, Y. (2005). Learning a similarity metric discriminatively, with application to face verification. In *CVPR*, pages 539–546.

Hadsell, R., Chopra, S., and LeCun, Y. (2006). Dimensionality reduction by learning an invariant mapping. In *CVPR*, pages 1735–1742. IEEE Comp. Society.

Hinton, G. and Salakhutdinov, R. (2006). Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504 – 507.

Hou, Y., Li, Z., Wang, P., and Li, W. (2018). Skeleton optical spectra-based action recognition using convolutional neural networks. *IEEE Trans. CSVT*, 28(3):807–811.

Hu, J., Zheng, W., Lai, J., and Zhang, J. (2015). Jointly learning heterogeneous features for RGB-D activity recognition. In *CVPR*, pages 5344–5352.

Hu, J., Zheng, W., Ma, L., Wang, G., Lai, J., and Zhang, J. (2019). Early action prediction by soft regression. *IEEE Trans. PAMI*, 41(11):2568–2583.

Koch, G., Zemel, R., and Salakhutdinov, R. (2015). Siamese neural networks for one-shot image recognition. In *ICML Deep Learning Workshop. vol. 2*.

Li, W., Zhang, Z., and Liu, Z. (2010). Action recognition based on a bag of 3D points. In *CVPR Workshops*, pages 9–14.

Liang, B. and Zheng, L. (2015). A survey on human action recognition using depth sensors. In *Int. Conf. on Digital Image Comp.: Techn. and Appl.*, pages 1–8.

Masci, J., Meier, U., Cireşan, D., and Schmidhuber, J. (2011). Stacked convolutional auto-encoders for hierarchical feature extraction. In *ICANN, vol. I*, pages 52–59.

Ren, B., Liu, M., Ding, R., and Liu, H. (2020). A survey on 3D skeleton-based action recognition using learning method. *arXiv, 2002.05907*.

Wang, L., Huynh, D. Q., and Koniusz, P. (2020). A comparative review of recent Kinect-based action recognition algorithms. *IEEE Trans. Image Process.*, 29:15–28.

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

Wang, P., Li, W., Li, C., and Hou, Y. (2018). Action recognition based on joint trajectory maps with convolutional neural networks. *Knowledge-Based Syst.*, 158:43 – 53.

Wang, P., Wang, S., Gao, Z., Hou, Y., and Li, W. (2017). Structured images for RGB-D action recognition. In *ICCV Workshops*, pages 1005–1014.

Wang, X., Hu, J.-F., Lai, J.-H., Zhang, J., and Zheng, W.-S. (2019). Progressive teacher-student learning for early action prediction. In *CVPR*, pages 3551–3560.

Wu, Y. (2012). Mining actionlet ensemble for action recognition with depth cameras. In *CVPR*, pages 1290–1297.

Xia, L. and Aggarwal, J. (2013). Spatio-temporal depth cuboid similarity feature for activity recognition using depth camera. In *CVPR*, pages 2834–2841.

Xia, L., Chen, C.-C., and Aggarwal, J. (2012). View invariant human action recognition using histograms of 3D joints. In *CVPR Workshops*, pages 20–27.

Yang, X., Zhang, C., and Tian, Y. L. (2012). Recognizing actions using depth motion maps-based histograms of oriented gradients. In *Proc. of the 20th ACM Int. Conf. on Multimedia*, pages 1057–1060. ACM.

Zheng, Y., Liu, Q., Chen, E., Ge, Y., and Zhao, J. L. (2014). Time series classification using multi-channels deep convolutional neural networks. In *Web-Age Information Management*, pages 298–310. Springer.

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