Continuous Hand Gesture Recognition for Human-Robot Collaborative Assembly

Bogdan Kwolek
AGH University of Krakow, 30 Mickiewicza Av., 30-059 Krakow, Poland
bkw@agh.edu.pl

Abstract

In this work, we present a framework for dynamic hand gesture recognition on RGB images acquired by an overhead camera. The recognition is realized for Methods Time Measurement-based planning of human-robot collaborative workspace. The 3D hand posture is estimated by MediaPipe. The recognition is done by a neural network in which a layer-wise feature combination takes place. We combine features extracted by basic blocks of Spatio-Temporal Adaptive Graph Convolutional Neural Network and by basic spatio-temporal self-attention blocks. We recorded and manually annotated 12 videos consisting of 54,659 RGB images with five basic motion sequences: grasp, move, position, release, and reach. We demonstrate experimentally that results of our networks are superior to results achieved by RNNs, ST-GCN, ST-AGCN, and CTR-GCN networks.

1. Introduction

Recognition of hand gestures is a very important problem due to numerous potential practical applications, including robotics [24], medicine, augmented reality [27], and virtual reality [26, 19, 13]. In recent years, there has been a growing interest in the problem of recognizing hand gestures [3, 1, 11, 30]. Li et al. [19] argue that hand gestures can be distinguished according to different spatiotemporal operational behaviors, modes of interaction, semantics, and ranges of interaction. Due to the significant development of sensor technologies on the one hand, and on the other hand due to the significant improvement in the ratio of computing power to the energy consumed for embedded systems, more and more attention is now being paid to techniques of recognizing dynamic gestures [26]. Dynamic gesture recognition techniques based on a single RGB camera, RGB-D sensors or multi-camera systems are among the most attractive due to the fact that they do not absorb the user's attention to a large extent [3].

Due to the rapid technological progress resulting in enhanced capabilities of service and collaborative robots, there is currently a significant increase in interest in the use of gesture recognition for the purposes of broadly understood robotics [20, 27, 24]. One of the most useful forms of human-robot interaction (HRI) is the collaborative assembly [24]. In these types of tasks, the user assembles a more complex object from certain parts through a sequence of sub-processes, where the collaborative robot actively assists the worker in completing the tasks. A person and a collaborative robot work on the same task, in the same workspace, at the same time. Sharing the same workspace is a fundamental element of this kind of HRI. To date, only a few vision-based systems have been developed to support human-machine collaboration by visual tracking the human posture or skeleton, recognizing hand gestures, detecting gazes and recognizing intentions [24].

A research carried out at Toyota Research Institute Europe resulted in an advanced neural model [29] that utilizes an artificial cognitive architecture to read the intentions of the human partner, using social cues to differentiate goals. The model was verified and validated in interactive HRI experiments, which consisted in jointly playing by a human and a collaborative robot arm a joint manipulation game using game blocks. In the block placing experiments the robot was able to correctly predict the partner's intentions using lightweight machine learning methods. Gao at al. [14] utilized a weighted sum fusion to combine the RGB, depth and 3D skeleton data in a framework developed for dynamic hand gesture recognition for human–robot interaction. The 3D hand skeleton have been determined by OpenPose. A 3CDNN+ConvLSTM neural network has been used to identify and classify the combined data with dynamic hand gesture. A neuro-inspired model for action selection in a human-robot join action scenario using Dynamic Neural Fields has been proposed in [9]. The model has been evaluated in a real construction scenario with the robot Sawyer, which selected the next part to be mounted, together with its human partner. The two-dimensional Action Execution Layer allows the representation of the components object and action in the same field. A recently published work [18] proposes an approach that extends the action recognition to multi-variant assembly processes. It uses generalized action primitives derived from Methods-Time Measurement (MTM) analysis, which are recognized on the basis of 3D skeletal data and a spatial-temporal graph convolutional neural network (ST-GCN). The 3D hand skeleton of has been extracted by the Kinect sensor.

In this work, we present a system for continuous hand gesture recognition for human-robot collaborative assembly. The working area that is shared between a worker and a robot is observed by an overhead RGB camera. The 3D hand posture is estimated by the MediaPipe. We propose a neural architecture that combines features extracted by basic blocks of adaptive graph neural network with features extracted by basic spatio-temporal self-attention blocks. We recorded and manually annotated 12 videos consisting of 54,659 RGB images with five basic motion sequences: grasp, move, position, release, and reach. We demonstrate experimentally that our network achieves superior results to results achieved by RNNs, ST-GCN, ST-AGCN, and CTR-GCN networks. The data for training and evaluating neural networks are available at: https://home.agh.edu.pl/~bkw/data/ACVR/.

The rest of the paper is organized as follows. In the next Section we discuss relevant work. Next, in Section 3 we outline preliminaries. We outline RNNs (Recurrent Neural Networks), ST-GCN (Spatio-Temporal Graph Convolutional Network), ST-AGCN (Spatio-Temporal Adaptive Graph Convolutional Network), and multi-head attention mechanisms. In Section 4 we present the proposed approach. In Section 5, after outlining our dataset, we present experimental results. In Section 6 we present conclusions.

2. Relevant Works

Human action recognition in RGB or RGB-D images is an active research area [32]. For 3D skeleton-based human action recognition several effective solutions relying on both graph neural networks [33, 25, 7] and transformer-based networks have been developed [31]. One of the important factors stimulating the development of skeleton-based action recognition algorithms was capability of the Kinect SDK of the real-time 3D skeleton estimation. Thanks to this a number of benchmark RGB-D datasets [21] were registered on which very good results of recognizing human actions were then obtained. In general, graph-based neural networks provide better results than networks based on transformer mechanisms [31]. However, in the near future we should expect a significant improvement of transformer-based skeleton action recognition.

Due to the fact that 3D hand skeleton estimation requires specialized equipment such as special gloves or multi-camera mocap systems, so far relatively few papers have been devoted to recognition of dynamic hand gestures based on 3D hand data. Most of work in this research area concerns dynamic gesture for sign language recognition or, ultimately, quite broadly understood human-machine interaction. A recently published CNN-based model for human-machine interaction [5] achieved high accuracy on recognition of five static gestures. A method for dynamic gesture recognition by combining 2D convolutional neural networks with feature fusion is proposed in [34]. In this method, the original keyframes and optical flow keyframes are utilized to extract spatial and temporal features, which are fed to a 2D CNN responsible for fusing them and final recognition. Recently, an effective deep architecture to classify in real-time various gestures from continuous data streams acquired by a live camera has been proposed in [2]. In a recent work, [4], Long-Term Short Memory (LSTM), Temporal Convolution Networks (TCN) and transformer-based models were proposed for isolated sign language recognition. The method employs MediaPipe [22] human pose estimator to estimate hand and face keypoints. It has been show experimentally that combining hand and face keypoints leads to improved recognition accuracy compared with networks operating on only hand keypoints.

3. Preliminaries

3.1. MediaPipe

Estimating hand posture (3D skeleton) based on RGB images is an inherently ill-posed problem due to the lack of depth information in the 2D input data. MediaPipe Hands is an efficient and accurate hand and finger tracking platform [22]. The discussed platform uses advanced artificial intelligence techniques to infer the 3D position of the characteristic points of the hand joints based on a single RGB image. It employs a single-shot palm detection model (SSD) and once this is done it carries out precise localization of 21 3D palm coordinates in the determined hand region. MediaPipe delivers several customizable pre-trained models to estimate and to track 3D hand skeletons. In this work, MediaPipe Hands is utilized to provide streams of 3D joints locations from RGB image sequences.

3.2. Recurrent Neural Networks

The LSTM (Long Short-Term Memory) [17] is a type of recurrent neural network that learns long-term dependencies between time steps of data stream. The LSTM has three gates: input, forget, and output, which are responsible for regulating the flow of information through the network. The input gate is responsible for deciding which information should let in, the forget gate decides which information to keep or discard from the cell state, whereas the output one decides which information to output. Each LSTM has a cell state through which the information is carried to the gates. The BiLSTM [15] builds and trains two LSTM neural networks operating together in the forward and backward directions. In contrast to the LSTM, GRU [8] has only two gates: reset and update, which makes it simpler and faster, but also less powerful and adaptable. The reset gate decides which information to discard from the previous hidden state, whereas the update gate decides which information should be added to the new hidden state. Generally, GRU neural networks outperform LSTMs on low-complexity data sequences while on high-complexity data sequences the LSTMs perform better.

3.3. Spatio-Temporal Adaptive Graph Convolutional Neural Network

Although RNN-based methods achieve good performance on multivariate time-series classification, modeling the skeleton data as sequence of joint coordinates is not an optimal solution, since the skeleton data is naturally embedded in a graph structure. Graph neural networks are well suited for capturing geometrical relations between joints due to their natural capability of handling non-euclidean data. ST-GCN [33] captures both patterns embedded in the spatial configuration as well as the temporal dynamics in skeleton sequences. Most of the skeleton-based action recognition approaches are built upon the base block of ST-GCN in which alternating between graph convolution and temporal convolution takes place. Spatio-temporal Adaptive Graph Convolutional Network (ST-AGCN) [25] has an attention mechanism and is capable of dealing with adaptive graphs. In contrast to ST-GCN networks the graph topology in this network is an optimized parameter and is unique for every layer. Additionally, owing to a residual branch it has better training stability than ST-GCN. ST-AGCN and its dual-stream (bone and joint) extension 2s-AGCN have been developed to recognize human actions from skeletal data streams. They are widely used for recognition of human actions on the basis of skeletons extracted by Kinect sensors, which deliver skeletons consisting of 25 joints.

The adaptive spatial graph convolution layer employs both the provided adjacency matrix as well as parameterized and optimized adjacency matrices. The adaptive spatial graph convolution can be described in the following manner:

$$f_{out} = \sum_{k}^{Kv} W_k f_{in}(A_k + B_k + C_k) \tag{1}$$

where $K_v = 3$ stands for the kernel size of the spatial dimension, W_k denotes the $d_{out} \times d_{in} \times 1 \times 1$ weight vector of the 1×1 convolution, where d is number of in/out channels, f_{in} is the input feature vector, f_{out} is the output feature vector, whereas A_k, B_k, C_k are the adjacency matrices. A_k represents the physical structure of the human hand in the form of the adjacency matrix, i.e. it determines whether there are connections between two graph vertexes. B_k is a matrix whose elements are parameterized and optimized along with other parameters when training the network. It allows learning new vertex connections. C_k is a data dependent adjacency matrix, which learns a unique graph for each sample. It is determined through embedding the input features through a 1×1 convolutional operation and a softmax function. If the number of input channels differs from the number of output channels, a 1×1 convolution is included in the residual path. The convolution for the temporal dimension is identical to the convolution in ST-GCN [33]. The spatial and temporal GCNs are followed by a batch normalization layer and a ReLU layer. The basic AGCN block is parameterized by: number of input channels d_{in}, number of output channels d_{out}, and stride s. If the number of samples is being halved, the number of output channels is doubled.

3.4. Multi-head Attention

Recently, a number of works have shown that transformer-based architectures such as Vision Transformer (ViT) [10] match or even surpass best CNN networks in image classification tasks, including ResNets [16]. The basic block of such architectures is the self-attention mechanism [28], which can learn the global dependencies between the input elements of data sequences. It performs sequence-to-sequence transformation where a sequence of vectors is fed on the input, and a sequence of vectors comes out at the output. One of the most important properties of the self-attention mechanism is the capability of adaptive adjusting relationships with neighbors according to their responses. Through determining a weighted average of sequence elements with the weights adaptively calculated on the basis of an input query and elements' keys, this mechanism dynamically decides on which elements is worth to attend more than others.

Given a set of inputs $X \in \mathbb{R}^{n \times d}$, and learnable parameter matrixes $W_q \in \mathbb{R}^{d \times d_q}$, $W_k \in \mathbb{R}^{d \times d_k}$, $W_v \in \mathbb{R}^{d \times d_v}$, where n is the sequence length, and d_q, d_k and d_v are the hidden dimensionality for queries/keys and values, respectively, the $query(Q = XW_q)$, $key(K = XW_k)$, and $value(V = XW_v)$ matrices are calculated first. Assuming that $d_q = d_k$, Q and K are of size $n \times d_k$, whereas V is of size $n \times d_v$, the softmax dot product self-attention is calculated as follows:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right) V \tag{2}$$

The dot product for every possible pair of queries and keys is performed in the matrix multiplication QK^T. This means that using the dot product as the similarity metric, the attention value from element i to j is based on its similarity of the query Q_i and the key K_j. The result of dot product is a matrix of size $n \times n$, where each row represents the attention logits for a specific element i to all other elements in the sequence. Under assumption that q and k are d_k–dimensional vectors with components that are independent random variables such their mean is equal to 0 and variance is equal to 1, then their dot product $q \cdot k = \sum_{i-1}^{d_k} u_i v_i$ has mean equal to 0 and variance equal to d_k. Hence, division by $\sqrt{d_k}$ should be performed to obtain the preferred dot product with the unit variance. Because of the above the result of dot product of the queries with the keys in (2) is scaled by the square root of d_k. The resulting attention scores are then fed into the softmax function. The attention weights calculated in such a way are used to scale the values V through a weighted multiplication operation.

One of the core mechanisms related to self-attention is the multi-head attention. It was introduced due to the observation that different elements of the sequence relate to each other in different ways. The query, key, and value matrices are transformed into h sub-queries, sub-keys, and sub-values, respectively, which are then processed through the scaled dot product attention independently. The independent attention outcomes are concatenated and then linearly transformed into the expected dimension. It is worth noting that the multi-head attention is permutation-equivariant with respect to its inputs.

4. Framework for Skeleton-Based Gesture Recognition

4.1. Methods-Time Measurement

Assembly time is one of the main estimates of assembly cost. Whether an assembly system is being planned or measures are being taken to increase the capacity of an existing assembly line, a fast and reliable method of estimating the time needed to complete a given assembly task is essential. Assembly time is defined as the time from start to finish assembly operation. Methods-Time Measurement is employed in industry to describe, analyze, evaluate and schedule manual tasks [12]. The MTM-1 standard considers five basic motions: grasping, moving, positioning, releasing, and reaching. Reach is a basic element of movement related to the movement of the hand or fingers. It is utilized to describe the movement of a hand or finger to a new destination. The grasp operation is employed to describe the control of one or more objects with the fingers or hands. The move operation is utilized to describe the phase of relocating the object to a new location. The positioning operation is usually preceded by the moving motion and is used to describe the orientation or positioning of one object relative to another object. Release is for describing the phase that ends control of an object with the hand or fingers. Planning a workplace based on the MTM is a time-consuming task, because each movement and its duration are usually determined by planning and ergonomics specialists. Due to the above, there is a fairly high demand for methods that automate such processes.

4.2. Scenario

The layout of the scene has been designed in such a way as to faithfully recreate the conditions that the vision system would have to deal in a situation with a real human-robot cooperative assembly. The worker's task is to pick up object by object from the box, move it to the stand the right side, and positioning it in the required pose and desired location. The scene is observed by a single overhead camera. The aim of the vision system is to continuously classify the hand motions into five classes. In a future work the Franka-Emika robot will employ a depth camera (Asus Xtion) that is already mounted on the gripper to grasp and then move the objects to new locations.

4.3. Proposed Approach

Let $f : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}$ be a function taking two vectors $\mathbb{R}^d$ as the input, then the self-attention score (weight) matrix S is defined as $S_{ij} = f(Q_i, K_j)\ \forall i, j \in [n]$. Let $h : \mathbb{R}^{n \times n} \rightarrow \mathbb{R}^{n \times n}$ be a normalization function, then the self-attention can be expressed in the following manner:

$$\text{Attention}(Q, K, V) = h(S) \cdot V \tag{3}$$

Hence, we assign higher weight to those value vectors whose corresponding key is most similar to the query.

Original transformer operates on sequential data, i.e., it is fed with a matrix $X \in \mathbb{R}^{n \times d}$. For sequence of skeletal data the input is a 3-order tensor $X \in \mathbb{R}^{n \times t \times d}$, where t stands for the number of skeletal data. This means that the input skeletal data has two dimensions, i.e., space and time. Thus, as in ST-GCN that alternates between graph convolution and temporal convolution, in each basic layer the data will be reshaped according to the following scheme: (i) reshape input to size $n \times td$, (ii) execute spatial attention, (iii) reshape $n \times td$ to $t \times nd$, (iv) execute temporal attention. The whole STSA-Net network consists of l such basic layers, usually $l = 8$. Depending on the temporal or spatial dimension, a spatial or temporal position features, are added to the data, which are then fed to Conv2D with 1×1 kernel and (s, 1) stride. As the position encoding functions the sine and cosine functions with different frequencies [28] are employed at this stage. In spatial encoding, the joints are encoded in turn, one after the other, and all frames in the sequence have identical encoding. In temporal encoding, a given joint is encoded over time sequentially. Positionally encoded and embedded queries/keys are utilized to compute the attention weights as in [28]. They are then used to scale the values through a weighted multiplication operation. The outcomes of all heads are concatenated and mapped by an linear layer to the output space $\mathbb{R}^{n \times t d_{out}}$. The output of the last layer is fed to a classification block. The stride s is equal to one for spatial dimension. For the temporal attention it is set to two, if it is necessary to reduce the number of samples by half. The basic STSA-Net block is parameterized by: number of input channels d_{in}, number of output channels d_{out}, number of internal channels d_k, and stride s. If the number of samples is being halved, the number of internal and output channels are doubled.

We investigated an architecture in which outputs of basic AGCN layers were concatenated with outputs of basic STSA-Net layers. By setting s to the same value in both concatenated networks, the outputs of layers to be concatenated have the same shape. As the concatenating the outputs of both networks is over the channel dimension, the number of concatenated channels is doubled. The concatenated output is fed to 1×1 Conv2d with a number of output channels two times smaller than the number of input channels.

5. Experimental Results

5.1. Dataset

A Logitech HD Pro Webcam C920 webcam was employed to acquire the images. It has been placed over the scene to observe the assembly tasks. The training set consists of 12 videos with a total of 54,659 RGB images of size 640×480. Figure 1 depicts sample images from the dataset with the hand actions considered in this work: reaching, grabbing, moving, positioning and releasing. All operations performed during dataset acquisition consisted only of the above-mentioned activities, which were always performed in the same order. All images that make up the training set were manually labeled, i.e. one of the five classes of MTM-1 movements was assigned to each frame. Images in which the hand was absent or a significant part of the hand was outside the image were marked as class six. The number of images in each category is as follows: reach – 3,553, grasp – 16,414, move – 5,621, position – 10,669, and release – 17,030. Five performers participated in the recordings of the training set. The test set consists of three movies with a total of 31,190 RGB images. The number of images in each category is as follows: reach – 3,911, grasp – 10,140, move – 5,523, position – 4,044, and release – 7,038. The actions were performed by two people who did not participate in the registration of the testing set. Determination of the 3D hand skeleton was carried out based on the MediaPipe Hands Python API. 21 points with 3D coordinates $x-$, $y-$ and $z-$ were determined on each image with the hand by the MediaPipe. Figure 2 shows the hand skeletons that have been determined by MediaPipe in the images from Fig. 1. After manually labeling the images, the 3D coordinates of the hand keypoints along with the action class were recorded for each image. Hand joint coordinates with class labels were stored in the train and the test datasets.

5.2. Experimental Evaluation

At the beginning, we implemented and trained recursive neural networks, which are commonly used to classify multidimensional time series. The neural networks were trained and evaluated on streams of 3D locations of hand joints, which were determined in advance by the MediaPipe. First, we trained an LSTM neural network. The network consisted of 128 hidden nodes in two layers, a fully connected layer with the number of neurons equal to half of the number of hidden nodes, and dropout layer followed by a classification layer with six neurons on the output. The size of the time-window was set to 32. After training the LSTM we determined its classification performance on the test subset. The accuracy, precision, recall and F1-scores achieved by this network are presented in the first row of Table 1. We calculated the macro-F1 score because we considered all classes to be equally important even though the test set is unbalanced. The number of hidden nodes as well as number of neurons in the fully connected layers for both networks were the same as in the LSTM network. Comparing the results obtained by these three mentioned networks, which classify multidimensional time series without taking into account the information resulting from the skeleton of the hand, it can be seen that the best results were obtained by the GRU neural network.

Next, we implemented, trained and evaluated an AGCN neural network. The network consists of seven basic TAGCN blocks with increasing output feature dimensions. It operates on input tensor of size (32,21,3), i.e. (time steps, joints, channels), which after reshaping it to (63,32) is fed to batch normalization layer. The output of batch normalization layer is reshaped to (3,32,21). The last TAGCN block is followed by a global average pooling layer, which is in turn followed by a fully connected layer on the output. Every basic TAGCN block is composed of a spatial graph convolution that is followed by a temporal graph convolution with respective kernel sizes. The output sizes of TAGCN layers are as follows: (64,32,21), (64,32,21), (64,32,21), (64,32,21), (128,16,21), (128,16,21), and (128,8,21). The temporal stride was set to 1 except the fourth and sixth layers for which the stride was set to 2. The global average pooling layer operates on tensors of size (1,128). The number of neurons in the last layer is equal to six. In the hand graph we included additional links of the fingertips to the base of the right neighbor finger, i.e. we additionally linked the following graph edges: 4-5, 8-9, 12-13, 16-17, c.f. MediaPipe hand skeleton. As can be seen in the fifth row of Table 1, the results achieved by the discussed neural network are better in comparison to results achieved by recurrent neural networks. We trained and evaluated also a ST-GCN neural network. The basic TAGCN block was replaced by TGCN block, and the same block parameters have been used to make the comparison of the classification performance fair. As can bee seen in Table 1 the results obtained by ST-GCN are not only worse than the results obtained by AGCN, but they are also worse than the results obtained by the GRU. In experimental evaluations devoted to GCN-based gesture recognition we trained and evaluated the recently proposed CTR-GCN [7] on our dataset as it achieved competitive results on action recognition benchmarks. This network leverages novel channel-wise topology refinement graph convolution (CTR-GC) to dynamically learn different topologies and effectively aggregate features in unlike channels. Significantly worse results than those obtained by the ST-GCT and AGCN may indicate that the recognition of hand gestures on 3D hand skeleton poses other challenges to graph-based networks than recognition of human skeleton-based action recognition, e.g. due to different articulation and movements.

Next, we implemented the STSA-Net. We trained several neural networks with different parameters, which we selected experimentally. Similarly as the AGCN, the STSA-Net operates on input tensor of size (32,21,3), which after reshaping to (3,32,21) is fed to input layer, that outputs tensors of shape (64, 32, 21). The output sizes of basic STSA-

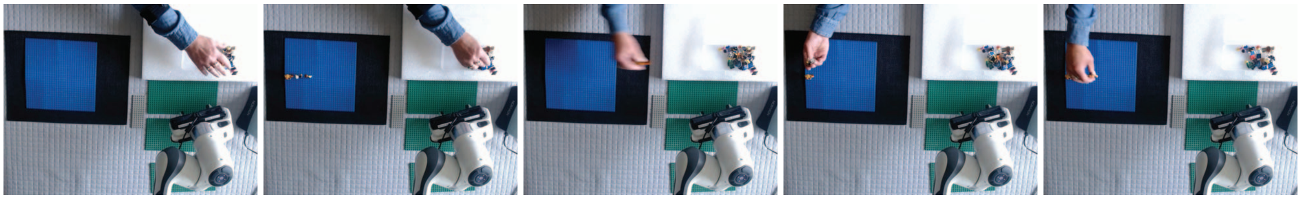

Figure 1. Five basic hand motions according to MTM: reach, grasp, move, position, and release (from left to right).

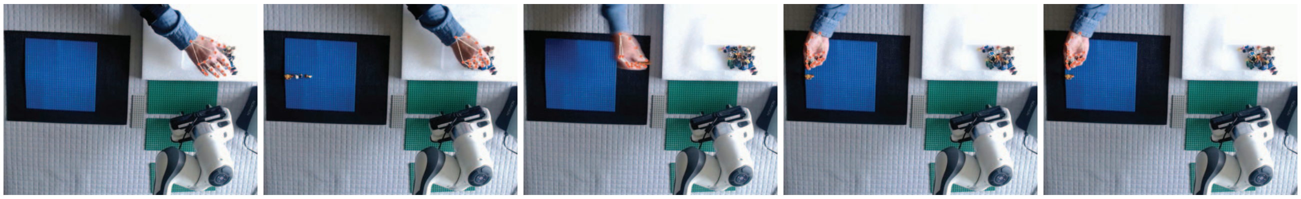

Figure 2. 3D hand skeletons calculated by the MediaPipe on images from Fig. 1.

Table 1. Accuracy, precision, recall and macro F1-score achieved by neural networks on our dataset for dynamic hand gesture recognition.

	Accuracy [%]	Precision [%]	Recall [%]	F1-score [%]
LSTM [17]	83.03	79.21	79.28	77.02
Bi-LSTM [15]	86.37	88.99	83.63	85.40
GRU [8]	87.01	88.65	84.19	86.01
ST-GCN [33]	86.78	88.95	82.93	85.22
AGCN [25]	89.46	90.73	86.61	88.23
CTR-GCN [7]	85.93	88.28	82.34	84.67
STSA-Net	90.49	90.89	88.43	89.56
AGCN-STSA-Net	91.39	90.69	90.03	90.31

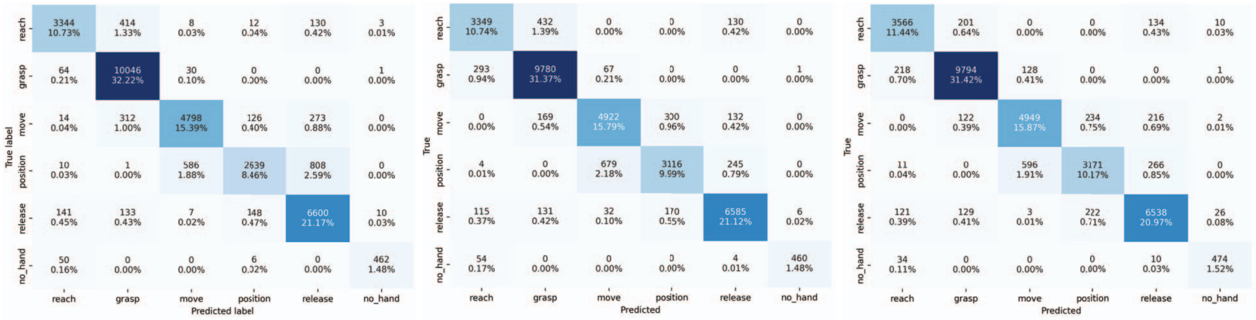

Figure 3. Confusion matrix on predictions of the AGCN, STSA-Net, and AGCN-STSA-Net networks.

Net layers are as follows: (64, 32, 21), (64, 32, 21), (64, 32, 21), (128, 16, 21), (128, 16, 21), (256, 8, 21), (256, 8, 21), and (256, 8, 21). The temporal stride was set to 1 except the fourth and sixth layers for which the stride was set to 2. The classification block was the same as in the AGCN. As we can observe in Tab. 1 the classification accuracy achieved by STSA-Net is far better than AGCN accuracy.

Finally, we implemented, trained and evaluated AGCN-STSA-Net network. The output of the first AGCN layer and output of the first STSA-Net layer have been concatenated over the channel dimension. Then the number of channels was reduced twice using Conv2D with kernel of size 1×1. Next, output concatenation over channel dimension and then channel reduction using Conv2D with kernel of size 1×1 has been executed on the output of the second layer. The output calculated in such a way has been processed the same way as in the STSA-Net. As we can observe in the last row of Tab. 1, the classification accuracy obtained by this neural network is better by 1.93% than the accuracy obtained by the AGCN neural network, and is better by 0.9% than the accuracy obtained by the STSA-Net network. AGCN-STSA-Net achieves better F1-scores than AGCN and STSA-Net neural networks. Figure 3 depicts confusion matrices achieved by discussed neural networks.

The trained neural networks have been utilized in experiments consisting in continuous hand gesture recognition. By striding the window along time axis the 3-order tensors have been extracted and then fed to the trained neural network. Figure 4 demonstrates example print-screen of the output window with sliders representing the classifications scores. On the basis of continuous gesture recognition we determined the execution times of individual hand gestures along with statistics illustrating the discrepancies from the average times.

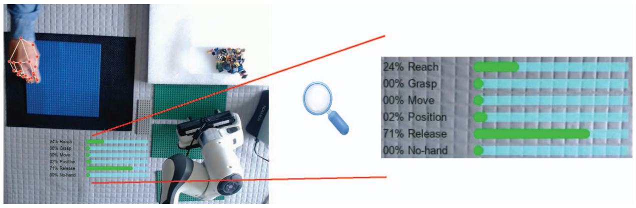

Figure 4. Output window with sliders representing the classification scores.

The neural networks were implemented in Python using PyTorch framework. They were trained in 25 epochs, with batch size set to 64. The training was performed using Adam optimizer and plateau learning rate scheduler [23] with an initial learning rate of 1e-4. Due to training networks on unbalanced datasets, as in [6], the AGCN, STSA-Net and our network were trained using the focal loss. The focal loss copes with class imbalance by down-weighting inliers (easy examples) such that their contribution to the total loss is small even if their amount is big. The size of the temporal window was set to 32 frames. The 3D joints streams were extracted using the MediaPipe with the minimum detection confidence set to 0.4, whereas the minimum tracking confidence for tracking the landmarks in consecutive images was set to 0.5. All networks were trained on two Nvidia A100 GPUs. Example video illustrating the continuous gesture recognition is available at: https://home.agh.edu.pl/~bkw/demos/AGCN-STSA-Net.avi, c.f. sample image on Fig. 4.

6. Conclusions

In this work we presented a framework continuous hand gesture recognition for human-robot collaborative assembly. We proposed a spatio-temporal self-attention network for dynamic gesture classification, and a spatio-temporal neural network that combines features extracted by graph-based layers with features extracted self-attention layers and then uses them for gesture classification. We recorded a dataset dynamic hand gesture recognition with manually labeled several thousand images. We demonstrated experimentally that the proposed neural networks have some potential. On our dataset they achieve better classification accuracies than ST-GCN, ST-AGCN, and recently proposed CTR-GCN.

References

[1] I.A. Adeyanju, O.O. Bello, and M.A. Adegboye. Machine learning methods for sign language recognition: A critical review and analysis. *Intell. Syst. with Appl.*, 12:200056, 2021. 1

[2] Apeksha Aggarwal, Nikhil Bhutani, Ritvik Kapur, Geetika Dhand, and Kavita Sheoran. Real-time hand gesture recognition using multiple deep learning architectures. *Signal, Image and Video Processing*, Jul 2023. 2

[3] Neena Aloysius and M. Geetha. Understanding vision-based continuous sign language recognition. *Multimedia Tools and Applications*, 79(31):22177–22209, 2020. 1

[4] Sarah Alyami, Hamzah Luqman, and Mohammad Hammoudeh. Isolated Arabic Sign Language recognition using a transformer-based model and landmark keypoints. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.*, 2023. 2

[5] U. Sai Babu, A. Raganna, K.N. Vidyasagar, S.H. Bharati, and Gautam Kumar. Highly accurate static hand gesture recognition model using deep convolutional neural network for human machine interaction. In *IEEE 4th Int. Conf. on Advances in Electronics, Comp. and Comm.*, pages 1–6, 2022. 2

[6] Danilo Barros Cardoso, Luiza C.B. Campos, and Erickson R. Nascimento. An action recognition approach with context and multiscale motion awareness. In *35th Conf. on Graphics, Patterns and Images*, volume 1, pages 73–78, 2022. 7

[7] Yuxin Chen, Ziqi Zhang, Chunfeng Yuan, Bing Li, Ying Deng, and Weiming Hu. Channel-wise topology refine-

ment graph convolution for skeleton-based action recognition. In *IEEE/CVF Int. Conf. on Computer Vision (ICCV)*, pages 13339–13348, 2021. 2, 6

[8] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *NIPS Workshop on Deep Learning*, 2014. 3, 6

[9] Ana Cunha, Flora Ferreira, Emanuel Sousa, Luis Louro, Paulo Vicente, Sergio Monteiro, Wolfram Erlhagen, and Estela Bicho. Towards collaborative robots as intelligent coworkers in human-robot joint tasks: what to do and who does it? In *52th Int. Symp. on Robotics*, pages 1–8, 2020. 1

[10] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *Int. Conf. on Learning Repr.*, 2021. 3

[11] El-Sayed El-Alfy and Hamzah Luqman. A comprehensive survey and taxonomy of sign language research. *Eng. Appl. of Artificial Intell.*, 114:105198, 2022. 1

[12] Gualtiero Fantoni, Salam Alzubaidi, Elena Coli, and Daniele Mazzei. Automating the process of method-time-measurement. *Int. J. of Productivity and Performance Management*, 12 2020. 4

[13] Tomohito Fujimoto, Takayuki Kawamura, Keiichi Zempo, and Sandra Puentes. First-person view hand posture estimation and fingerspelling recognition using HoloLens. In *IEEE 11th Global Conf. on Consumer Electr.*, pages 323–327, 2022. 1

[14] Qing Gao, Yongquan Chen, Zhaojie Ju, and Yi Liang. Dynamic hand gesture recognition based on 3D hand pose estimation for human–robot interaction. *IEEE Sensors Journal*, 22(18):17421–17430, 2022. 1

[15] Alex Graves, Navdeep Jaitly, and Abdel-rahman Mohamed. Hybrid speech recognition with deep bidirectional LSTM. In *IEEE Workshop on Automatic Speech Recognition and Understanding*, pages 273–278, 2013. 2, 6

[16] Kai Han, Yunhe Wang, Hanting Chen, Xinghao Chen, Jianyuan Guo, Zhenhua Liu, Yehui Tang, An Xiao, Chunjing Xu, Yixing Xu, Zhaohui Yang, Yiman Zhang, and Dacheng Tao. A survey on vision transformer. *IEEE Trans. on PAMI*, 45:87–110, 2023. 3

[17] Sepp Hochreiter and Jürgen Schmidhuber. Long Short-Term Memory. *Neural Comput.*, 9(8):1735–1780, 1997. 2, 6

[18] Julian Koch, Lukas Buesch, Martin Gomse, and Thorsten Schueppstuhl. A Methods-Time-Measurement based approach to enable action recognition for multi-variant assembly in human-robot collaboration. *Procedia CIRP*, 106:233–238, 2022. 2

[19] Yang Li, Jin Huang, Feng Tian, Hong-An Wang, and Guo-Zhong Dai. Gesture interaction in virtual reality. *Virtual Reality & Intelligent Hardware*, 1(1):84–112, 2019. 1

[20] Hongyi Liu and Lihui Wang. Gesture recognition for human-robot collaboration: A review. *Int. J. of Industrial Ergonomics*, 68:355–367, 2018. 1

[21] Alexandre Lopes, Roberto Souza, and Helio Pedrini. A survey on RGB-D datasets. *Computer Vision and Image Understanding*, 222:103489, 2022. 2

[22] Camillo Lugaresi, Jiuqiang Tang, Hadon Nash, Chris McClanahan, Esha Uboweja, Michael Hays, Fan Zhang, Chuo-Ling Chang, Ming Guang Yong, Juhyun Lee, Wan-Teh Chang, Wei Hua, Manfred Georg, and Matthias Grundmann. MediaPipe: A framework for building perception pipelines. *CoRR*, abs/1906.08172, 2019. 2

[23] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In *4th Int. Conf. on Learning Representations, ICLR*, 2016. 7

[24] Francesco Semeraro, Alexander Griffiths, and Angelo Cangelosi. Human–robot collaboration and machine learning: A systematic review of recent research. *Robotics and Computer-Integrated Manufacturing*, 79:102432, 2023. 1

[25] Lei Shi, Yifan Zhang, Jian Cheng, and Hanqing Lu. Two-stream adaptive graph convolutional networks for skeleton-based action recognition. In *IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 12018–12027, 2019. 2, 3, 6

[26] Yuanyuan SHI, Yunan LI, Xiaolong FU, M.I. Kaibin, and M.I. Qiguang. Review of dynamic gesture recognition. *Virtual Reality & Intelligent Hardware*, 3:183–206, 2021. 1

[27] Ryo Suzuki, Adnan Karim, Tian Xia, Hooman Hedayati, and Nicolai Marquardt. Augmented reality and robotics: A survey and taxonomy for AR-enhanced human-robot interaction and robotic interfaces. In *Proc. of the CHI Conf. on Human Factors in Computing Systems*, CHI '22. ACM, 2022. 1

[28] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proc. of the 31st Int. Conf. on Neural Information Processing Systems*, NIPS'17, pages 6000–6010. Curran Associates Inc., 2017. 3, 4

[29] Samuele Vinanzi, Angelo Cangelosi, and Christian Goerick. The role of social cues for goal disambiguation in human-robot cooperation. In *29th IEEE Int. Conf. on Robot and Human Interactive Comm. (RO-MAN)*, pages 971–977, 2020. 1

[30] Ankita Wadhawan and Parteek Kumar. Sign language recognition systems: A decade systematic literature review. *Archives of Comp. Methods in Eng.*, 28(3):785–813, 2021. 1

[31] Wentian Xin, Ruyi Liu, Yi Liu, Yu Chen, Wenxin Yu, and Qiguang Miao. Transformer for skeleton-based action recognition: A review of recent advances. *Neurocomputing*, 537:164–186, 2023. 2

[32] Santosh Kumar Yadav, Kamlesh Tiwari, Hari Mohan Pandey, and Shaik Ali Akbar. A review of multimodal human activity recognition with special emphasis on classification, applications, challenges and future directions. *Knowledge-Based Systems*, 223:106970, 2021. 2

[33] Sijie Yan, Yuanjun Xiong, and Dahua Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *Proc. of AAAI Conf. on Art. Intell.*, 2018. 2, 3, 6

[34] Jimin Yu, Maowei Qin, and Shangbo Zhou. Dynamic gesture recognition based on 2D convolutional neural network and feature fusion. *Scientific Reports*, 12(1):4345, 2022. 2