# Dynamic hand gesture recognition using Multi-Branch Attention Based Graph and General Deep Learning Model

# xzfdsfdsf, a) and Author's Name2, 3, b) (Use Times New Roman Font: 14 pt, Centered) (Names should be written in First Name Surname order)

Author Affiliations

(Use the Microsoft Word template style: Author Affiliation) or (Use Times New Roman Font: 10 pt, Italic, Centered)(if authors share the same affiliation, list the affiliation one time and number accordingly)

1Replace this text with an author's affiliation   
(use complete addresses, including country name or code).

2Additional affiliations should be indicated by   
superscript numbers 2, 3, etc., as shown above.

3You would list an author's second affiliation here.

Author Emails

Use superscript letters such as "a)" to indicate the author's e-mail address below.   
Use b), c), etc., to indicate e-mail addresses for more than one author.   
a) Corresponding author: your@emailaddress.xxxb) anotherauthor@thisaddress.yyy

**Abstract.** The dynamic hand skeleton data has become increasingly attractive to widely studied for recognising hand gestures that contain 3D coordinates of hand joints. Many researchers have been working to develop a skeleton-based hand gesture recognition system using various discriminative spatial-temporal attention features by calculating dependencies between the joints. However, those methods may be faced difficulties in achieving high performance and generalizability due to inefficient features. To overcome the challenges, we propose the Multi-Branch Attention Based Graph and General Deep Learning model to recognize hand gestures by extracting all possible kinds of skeleton-based features. We used two graph-based neural network channels in our multi-branch architectures and one general neural network channel. In graph-based neural network channels, one channel first used the spatial attention module and then the temporal attention module to produce the spatial-temporal features. On the other hand, using the reverse sequence of the first branch, we produced temporal-spatial features in the second channel. In the last branch, a general deep learning-based features was extracted by using a general deep neural network module. The final feature vector was constructed by concatenating spatial-temporal, temporal-spatial, and general features and feeding them into the fully connected layer. We included position embedding and mask operation for both spatial and temporal attention modules to track the sequence of the node and reduce the system's computational complexity. Our model achieved 94.12%, 92.00%, and 97.01% accuracy after being evaluated with MSRA, DHG, and SHREC'17 benchmark datasets respectively. The high-performance accuracy and low computational cost proved that the proposed method outperformed the existing state-of-the-art.

**Keywords:** Hand Gesture Recognition, Machine Learning, Spatial-Temporal Attention (STA), Deep Learning, Hand Skeleton.

# Introduction

Research on hand gesture recognition has been increasing daily since many real-life applications like human-computer interaction, nonverbal communication, controlling a wheelchair, abnormal behavior monitoring, and sign language recognition [1-6]. Previous work on hand gesture recognition has been divided into two categories based on the data collection procedure: vision-based and sensor-based systems. Since the sensor-based system is difficult because of the carrying sensor, researchers focus on the vision-based system because it only uses a camera, and that is easy to carry. Based on the input data modality, vision-based research work can be divided into two categories: image-based research, which uses full image pixels, and skeleton-based research, which uses only joint information. RGB or RGB-D images are common input for an image-based method for extracting the recognition features. In comparison, skeleton-based methods predict hand gestures based on 2D or 3D coordinates of hand joints. The skeleton sequence is not affected by the limitations of the RGB video and does not consist of color information. Moreover, Johansson et al. have proved that key joints of the gesture carry highly potential information about human motion [7]. Furthermore, each skeleton joint represents a point of the human body in three dimensions coordinates. Among the significant reasons this dataset is valuable to researchers is that it contains higher-level semantic information with a small amount of memory and adapts easily to dynamic systems [8-10]. Currently, many low-cost depth cameras are available in the market, like Intel RealSense, and Oak-D, which are easy to use for collecting skeleton gesture information and made great progress in gesture recognition research [11-12]. Based on the skeleton-based dataset, many researchers proposed conventional methods for designing a powerful feature descriptors model for recognizing hand gestures [13-15]. The main problems of the conventional approach are less performance accuracy and limitations of capabilities for generalization. Researchers applied deep learning techniques to overcome the challenges and improve performance accuracy by directly converting joint coordinates into tensors that feed neural networks [16, 17, 18]. They first produced the feature with the neural network, which is learned by the deep learning network for training. Some other researchers transformed their input skeleton into a meaningful format like a graph, a point sequence or a pseudo image using graph topologies or traversal rules. Then this data format is directly fed into the deep learning method such as CNN, GCN or RNN, LSTM to extract effective features for improving the network architecture performance [8,19,20]. Moreover, until now, there is some uncertainty about whether the hand-crafted extracted features and rules are the optimal choice of joint global dependencies for the model. However, learning global agencies transformer has produced success in the natural language processing (NLP) field, which mainly includes the self-attention mechanism [20,21,22,23]. They reported that better parallelizability and global dependency could be learned with minimum computational complexity among the element. In addition, the attention-based model does not require information about the intrinsic relationship among the joints. Another suitability of the attention model is that there is a limited number of joints in the hand gesture dataset. With minimum computational cost, it is possible to discover useful patterns from the hand skeleton dataset.

The main drawback of these models nevertheless considers the spatial and temporal structure of the sequential hand skeleton dataset. Recently, many researchers applied a graph-based spatial-temporal attention model to recognize skeleton-based hand gestures [24-29]. Although they achieved good performance, the main drawback is the lack of flixibility and sub-optimal performance because of the fixed graph structure which may be difficulty capturing variance and dynamics across different actions. To overcome the challenges, more recently researchers works to develop dynamic hand-skeleton dynamic graph based spatial-temporal model to recognize hand gesture [25, 30-31] . Also, they overcome the optimality problem with their model but their performance accuracy is not satisfactory. Moreover, their model may be faced difficulties in achieving satisfactory peformance or same performance all the time because of the inefficiency of the extracted feature. However, they only, extracted spatial feature then temporal feature and there is no explaination about the vice versa featues or if the combination of the other features. To overcome the challenges we are inspired to extract all posible kind of features from the hand skeleton dataset with the dynamic graph based attention model including spatial and temporal information.

To do this, sudy, we proposed Multi-Branch Attention Based Graph and General Deep Learning model for hand gesture recognition using a skeleton dataset to overcome the mentioned challenges. We developed the architecture by following the dynamic graph based attention-based mechanism including spatial, temporal and deep learning information. To convert from original nonsequential skeleton information to sequential information, we used a general neural network, considered the initial feature.

Then we employed three branches to extract all possible feature with spatial-temporal, temporal-spatial and general deep neural network branch. We considered the spatial-temporal, temporal-spatial branch as a graph-based deep neural network branch that used a position embedding technique to generate the unique markers for every point before each attention block, which helped the attention model feed data sequentially. We utilized masking operation in each attention blocks to reduce the computational complexity because two individual information would decrease the efficiency of the system. The main purpose of the third layer, to recover the missing feature value and solve inefficient signal propagation in the fully connected layer. As we fused the three kind of features, which combined all possible kind of features of the hand skeleton, it became an efficient and quicker process compared to the existing system. The significant contribution of this study is as follows:

* We propose a Multi-Branch Attention Based Graph and General Deep Learning Model to recognize dynamic skeleton-based hand gestures.
* We propose several principles in designing spatial-temporal, temporal-spatial, and general deep neural network models. The first branch produces spatial-temporal features based on spatial attention through a temporal attention block, and the second produces temporal-spatial features based on temporal attention through a spatial attention block. The third branch carries the general deep learning network features, and finally, we fused three features vector to generate the effective final feature vector.
* We conducted a comprehensive validation of our system with the three-dynamic skeleton-based hand gesture dataset and achieved superior performance over the state-of-art method considerably within minimum time. The models and code of the proposed model uploaded to GitHub to make public1.

This paper we organized as follows, relevant literature study and work provided in Section~\ref{sec2}. Section~\ref{sec3}~ is described the benchmark dataset of hand skeletons used to develop this work. The proposed multi-branch spatial-temporal attention model is described in Section~\ref{sec4}.

Section~\ref{sec5}~described the experimental results and different evaluation scenarios. Section~\ref{Sec6}~ concludes the paper, including some future work.

# 2. Related Work

**Hand joint skeleton information-based hand gesture recognition has recently been widely used in the computer vision domain but is still considered a challenging task.**

The traditional approach, like machine learning and traditional feature extraction method, mainly focuses on developing effective feature descriptors [15,32,33,34,35]. Ohn-Bar et al. l. proposed a set of feature generators from a skeleton dataset by including a histogram of oriented gradients (HOG) algorithm with the descriptor and employed linear SVM after converting the feature to a 2D array using HOG again [15]. Many other feature extractors have also been proposed by researchers, like the covariance matrix for skeleton joint location [35], joint location, joint angles, 3D geometric relationships between [36], and intraclass variance [37]. Hand geometric configuration for capturing hand shape variation was proposed by Smedt et al. which is used to extract spatiotemporal motion features of hand parts from the whole Euclidean space [38]. They achieved 82.50% and 80.11% accuracy for the 14 and 28 gestures of the DHG dataset after applying SVM machine learning on the Riemannian based trajectory features. Smedt et al. extracted features based on the fisher vectors and skeleton-based geometric technique, then applied SVM to the concatenated features, and achieved 83.00% and 80.00% accuracy for DHG dataset 14 gesture and 28 gesture sequentially [13]. They extracted three features, namely the shape of connected joints (SoCJ), histogram of hand directions (HoHD), and histogram of wrist rotations (HoWR) and combined them to make the final feature vector. Smedt et al. also applied the fisher vector and shaped the connected feature for the SHREC'17 dataset with the SVM classifier and achieved 88.24% and 81.90% for 14 gestures and 28 gestures sequentially [14]. The advantage of the work is that they demonstrated the superiority of 3D skeleton information over depth-based approaches, but the drawback is they did not consider the amplitude of the gesture and temporal pyramid representation may lose some information. Chen et al. proposed a motion feature extractor by combining articulated movement of the finger and motion feature from global hand movement for extracting bone angle and applying RNN for classification. They evaluated their model with the DHG dataset and achieved 84.68% and 80.32% accuracy for the 14 and 28 classes, respectively [16].

Also, some researchers employed deep neural networks like CNN on the hand joints skeleton data for recognizing hand gestures and significantly improved [14,18, 23, 16,28,33]. Many researchers used other networks with CNN, like an RNN-based approach that transforms the skeleton data into sequential data using traversal rules and feeds into LSTM for training and prediction [9,17,18,39,40]. Lin C et al. developed a fusion model by combining skeleton LSTM and Res-C3D network for recognizing abnormal hand gestures [41]. Lai et al. incorporated a CNN and an RNN deep learning model for recognizing skeleton-based hand gestures and achieved 85.61% for the DHG 14 gesture dataset [42]. Ma et al. employed an unscented kalman filter (UKF) to reduce the noise and include LSTM for classification [26]. They focused on the noisy dataset by revising the noise in the hand skeleton data and achieved 85.92% and 80.44% accuracy for the 14 and 28 gestures of the DHG dataset sequentially. Nunez et al. proposed a combination of CNN and LSTM model for recognizing a temporal 3D pose and they achieved 85.46% and 81.10% accuracy for the 14 and 28 gestures of the DHG dataset, respectively [15]. Chen et al. employed an augmented network based on motion (MFA-Net) for recognizing hand gestures, and they achieved 85.75% and 81.10% for the DHG dataset and 91.31 and 86.55% accuracy for the 14 and 28 gestures of the SHREC'17 dataset respectively [27]. They extracted features using a variational autoencoder from finger motion and global motion and then fed the features into 3 RNNs. Ma et al. proposed a modified memory-augmented neural al network, namely gesture recognition using an enhanced network (GREN) and LSTM architecture to recognize hand gestures as a short learning algorithm that aims to improve the system's efficiency [23]. They achieved 82.29% and 82.03% for the DHGD dataset and 79.17% for the MSRA dataset. Handwriting-inspired features (Hif3d) is proposed **by** Boulahia et al. for a 3D skeleton-based gesture classification and achieved 90.48% for 14 gestures and 80.48% accuracy for the 28 gestures of the DHG dataset [29]. Recently, researchers have focused on utilizing self-attention mechanisms to increase the efficiency and performance accuracy of the vision-based hand gesture recognition task by reducing the long-range distance [43-44]. Vaswani et al. first applied a self-attention network to establish a semantic relationship among words [21]. Query, Key and Values are the main elements of this method where multiply the Query with Key in the first stage, then divide by the dimension of the key and finally apply the SoftMax function to produce the weight vector [22,31]. After that, it is also employed for detection, semantic segmentation, and relational modelling research work [**45-47**]. Current day, many researchers combined spatial-temporal attention with various architectures like CNN [40 48, 49-51], RNN and soft-attention instead of hidden RNN [52], and memory attention networks (MANS) [32]. Song et al. applied a spatial-temporal attention mechanism through RNN and LSTM, where they used individual joints as the main information [53]. Hou et al. employed spatial-temporal attention by combining with residual connection and temporal convolutional neural network (STA-Res-TCN) to recognize skeleton based human gestures [28]. They extracted features from the different levels of attention mechanism and applied CNN for individual time steps and finally achieved 89.20% and 85.00% for 14 and 28 gestures of the DHG dataset, respectively. They also evaluated the model SHREC17 dataset and achieved 93.60% and 90.70% accuracy for 14 and 28 gestures sequentially. Recently, a graph convolutional neural network (GCNN) was used by many researchers for gesture recognition [8,30, 33,54]. Also, the existing system produced a good performance in some cases but still faced some generalisation problems and sometimes difficulties in achieving high performance for more datasets. To overcome the challenges, we employed here a Multi-Branch Attention Based Graph and General Deep Learning model to recognize hand gestures. We first employed a deep neural network and then employed a spatial-temporal and temporal-spatial branch to produce node and edge features for spatial and temporal domains. To increase the system's generalization, we also extracted general deep learning features and concatenated the extracted three features for producing final feature vector. To reduce the computational cost, we used a spatial-temporal mask and achieved 94.12% accuracy for the MSRA dataset, then 92.00% and 88.78% accuracy for the DHG dataset. In the same way, they achieved 97.01% and 92.78% accuracy for the 14 and 28 gestures sequentially for the SHREC'17 dataset. Our study is more efficient in general, as it does not require hand-crafted transformation rules, and it produced high performance compared to the existing method by a significant margin.

# Dataset

We studied nine open sources of the skeleton and handed gesture-based dataset to evaluate the model, namely: MSRA [55], DHG [13], SHREC17[14], Florence 3-D action [56], UTKinetic [57], UCF-Kinetic [58], NTU [59], NYU [60], ICVL [61], NVGesture[62] are considered as the benchmark dataset. Among them, Florence 3-D action, UTKinetic, UCF-Kinetic, and NTU datasets are human action dataset. NYU datasets are collected for only binary data, and NVGesture and ICVL contain only RGB, depth information. Since our objective of the proposed model is to recognize skeleton based hand gestures, we used the most recent skeleton-based hand gesture datasets namely: MSRA, DHG and SHREC17, which contained almost the same number of hand skeleton points. The details of the uses skeleton dataset are given below [5]:

3D Skelton data sequence can be defined as a vector according to Equation (1)

S (1)

Here, S represent the skeleton data sequence, represents a multivariate time sequence, T represent the transpose, and each component of the sequence we can be **written** as which contained three univariate sequence components like the following Equation (2).

(2)

Where represent the position of the -the skeletal joint and every joint contains a precise or distinct articulation of the hand of the physical world. From each t time frame, 21 joints for the MSRA dataset and 22 joints for the DHG and SHREC'17 dataset are collected in 3D space by Intel Creative Interactive Camera by considering the position , , where N=21 and 22.

**MSRA Dataset**

One of the hand joint skeleton-based gesture datasets is the MSRA, which is the most challenging publicly available sequence data [55]. This dataset was recorded from 9 participants based on 17 right-hand gestures using Intel Creative Interactive Camera. Each gesture is manually chosen by following the American sign language gesture focusing on the fingure articulation's span as much as possible. The dataset contains 490 to 500 frames for each gesture, and for 17 gestures, it is composed of 76500 frames.

There are 21 joints as 3D world space or skeleton information in each frame and also collected 2d depth images as well. Among the 21 joints, each finger consists of four joints and one in the palm for the MSRA dataset. The name of the 21 joints is shown in Figure 1. This dataset is considered challenging because of the viewpoint variation.

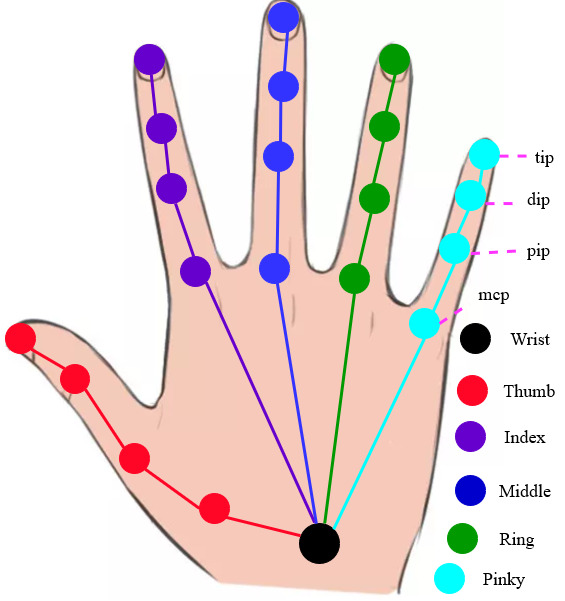


Figure 1: Twenty-one joints of MSRA dataset with right-hand skeleton

**DHG Dataset**

DHG is a publicly available dynamic and one of the challenging datasets for hand gestures, which contains a sequence of 14 right-hand gestures with various finger configurations [13]. For each gesture, the dataset was collected using Intel Real sense SDK and five times from 20 participants in two ways of finger configuration. By following the procedure, they collected a total of 2800 video sequences, and the length of each video contains 20 to 70 frames. Each frame is considered a 3D world space and a full hand skeleton formed with 22 skeleton joints. Some gestures consist of hand movements called coarse gestures, and some other gestures are composed of the shape of a hand, called fine gestures. Among the 14 gestures, nine coarse and five fine-grained gestures are reported. Also, the DHG dataset contains depth image skeleton information, but in our experiment, we used only the skeleton information for gesture recognition. Table 1 shows the name and types of the gestures. Figure 2 shows the name and position of the 22-hand skeleton.

|  |  |  |
| --- | --- | --- |
| **No.** | **Name of the Gesture and**  **Tag name** | **Gesture Type** |
| G-1 | Grab (G) | Fine |
| G-2 | Tap (T) | Coarse |
| G-3 | Expand (E) | Fine |
| G-4 | Pinch (P) | Fine |
| G-5 | Rotation clockwise (RC) | Fine |
| G-6 | Rotation counter-clockwise (RCC) | Fine |
| G-7 | Swipe right (SR) | Coarse |
| G-8 | Swipe left (SL) | Coarse |
| G-9 | Swipe up (SU) | Coarse |
| G-10 | Swipe down (SD) | Coarse |
| G-11 | Swipe x (SX) | Coarse |
| G-12 | Swipe + (S+) | Coarse |
| G-13 | Swipe v (SV) | Coarse |
| G-14 | Shake (Sh) | Coarse |

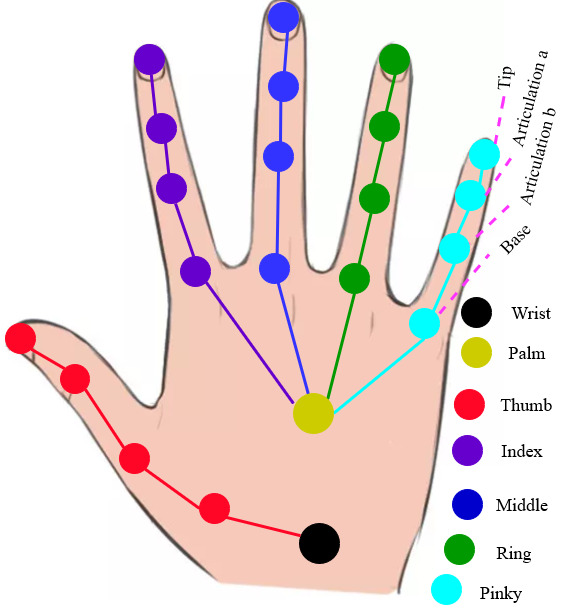


Figure 2: Twenty-two joints for the DHG and SHREC'17 dataset from the right-hand skeleton

**Table 1.** Name of the 14 gestures for the DHG and SHREC'17 dataset.

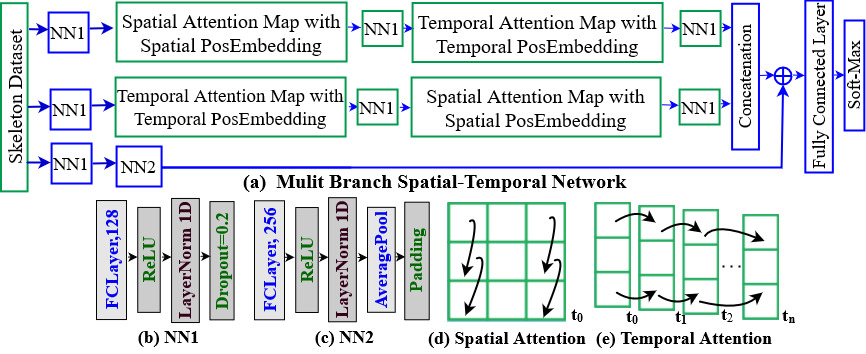
**SHREC'17 Dataset**

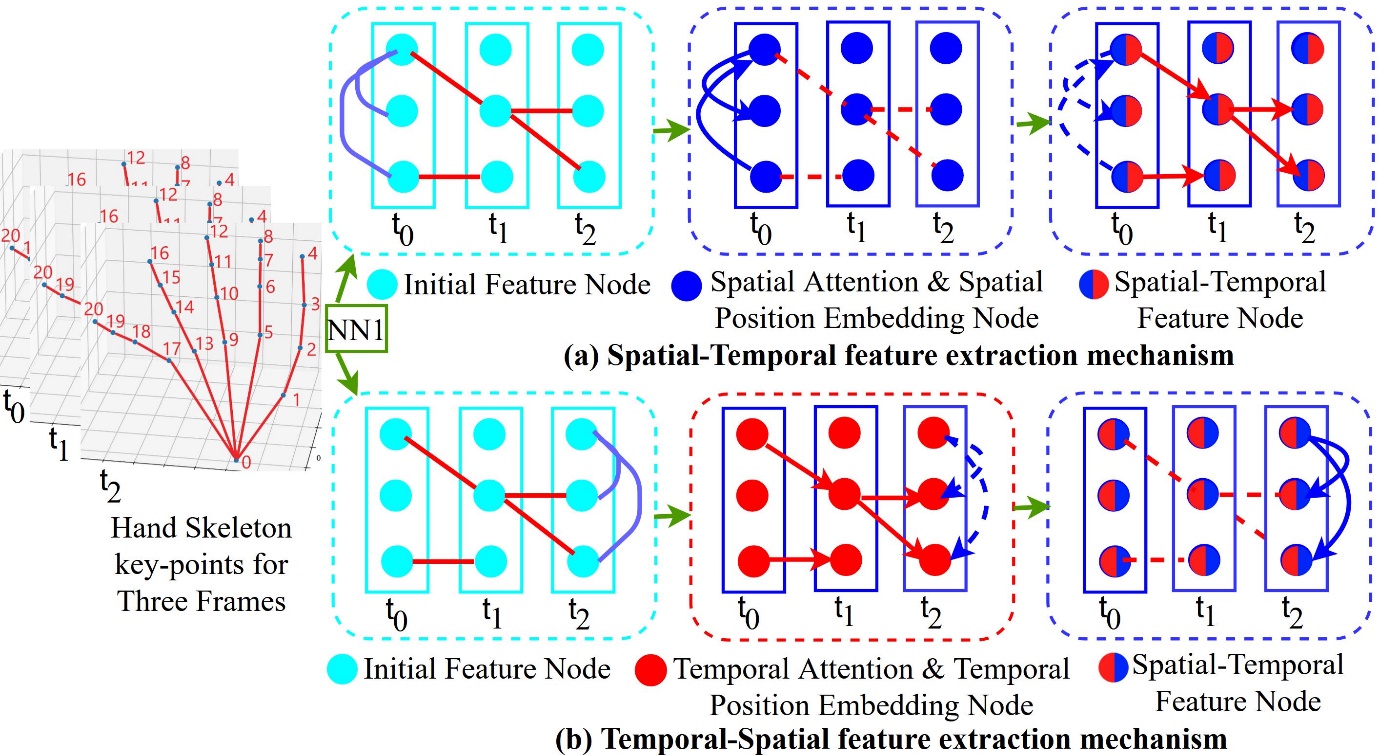
Another challenging skeleton-based hand gesture dataset name is the SHREC'17 dataset [14]. This dataset is the same as the DHG dataset, and the Intel Reals Sense SDK also used collected from 27 participants. Data were collected 1 to 10 times from each participant in a 2-finger configuration, with a total of 2800 video sequences. Depending on the number of fingers, labels from this dataset are categorized as 14 labels or 28 labels. In addition, among the gestures, some gesture consists of only one finger, and some gesture is composed of a whole hand. For each gesture, a 2D and 3D hand representation was also collected with the depth image for each scene and each time step. Although this dataset contains 2D depth images and 3D hand skeleton information, we used only 3D hand skeleton information in this study. The 22 hands skeleton points name of this dataset shown in Figure~\ref{fig:f2} and the name of the shown in Table~\ref{table1}.

**https://projet.liris.cnrs.fr/eg3dor17/#shrec**

# Methodology

The main goal of the demonstrated MSRA, DHG and SHREC'17 dataset was (1) full hand skeleton and depth information based dynamic hand gesture recognition, (2) evaluating the efficiency of the hand gesture recognizer based on the number of the finger in the gesture. However, the main objective of our study is not the same as theirs because our study is to achieve high performance in hand gesture recognition with minimum time and cost using only 3D hand skeleton information comparing the still image and video-related hand gesture recognizer. Another objective of our study is to propose a small deep neural network as a skip connection to resolve the missing value problem and improve the performance of hand gesture recognition. Our proposed study is demonstrated in Figure 3(a), where we parallelly employed two branches of attention block, namely spatial-temporal and temporal-spatial and concatenated the output of the two branch features. Each branch took input from the output of a neural network, namely NN1, shown in Figure 3(a). Firstly, NN1 takes input as a skeleton data point for each node and projects the input hand joints 3D coordinate into a primary feature node that is 128 dimensions. All three branches took initial features as input, where the first and second branches embedded the output of NN1 with associate spatial and temporal position to track the sequence correctly. The 1st branch firstly produces the spatial feature with 256 dimensions as a node feature and is projected into a 128-dimension using the neural network NN1, then embedded with temporal position and fed into the temporal attention model, which is produced the temporal feature with 256 dimensions. After that, we projected the 256-dimension temporal node feature into 128 using NN1, which is considered the spatial-temporal feature . Figure 4 (a) shows the Spatial-Temporal feature extraction mechanism. In the 2nd branch, we follow the reverse sequence of the 1st branch, where we first fed the initial feature into the temporal attention model and then fed the temporal feature into the spatial attention model and produced the temporal-spatial feature vector . Figure 4 (b) shows the Temporal-Spatial feature extraction mechanism. The 3rd branch also took the as input, and after applying the general deep neural network NN2 which is shown in Figure 3(c), it produced a general feature . Then, we concatenated the spatial-temporal, temporal-spatial and general features according to Equation 3 and produced the final feature vector of the proposed architecture . Lastly, we fed the average pool feature vector of the concatenated node features into the fully connected layer for classification.

 (3)

Figure 3: Proposed working Flow Architecture

## Figure 4: Spatial-Temporal and Temporal-Spatial feature extraction procedure.

## Graph-Based Deep Neural Network Branch

## We considered two among the three branches as the graph-based deep neural network branch because we used the attention-based mechanism for computing the representation of every joint node of the hand skeleton as a graph node by following its neighbors. The self-attention approach helps us learn an adaptive and dynamic local summary of the neighbor node to improve the prediction, then change to multi-head attention by repeating itself. Extracting a spatial-temporal and temporal-spatial domain feature is the primary purpose of these two branches to build a long sequence for learning the most important part of the hand skeleton. To modify the unified graph, we need to extract spatial and temporal domain features which is dynamically optimized by the different actions.

## Both graph-based branches took input from the output of NN1 and produced the spatial-temporal and temporal-spatial features after encoding with the spatial and temporal attention model. In both branches, we employed position embedding and masking operations for each attention at spatial and temporal domains to improve performance accuracy and efficiency.

### Graph Initialization with Skeleton

The structure of the hand skeleton data naturally looks like a graph when we consider it a graph. Given a hand gesture video sequence which contains T frames to represent the hand skeleton and the total N number of 3D hand skeleton joints can be recorded from each of the frames. We assumed, a fully connected graph is constructed from the sequence of hand skeleton joints of a frame which is considered as . Let the set of the node denoted by the of the graph and i-th hand joints of the time steps, t is contained by the node . The feature of the node is contained by the and feature of all nodes are written by . The main concept of the feature extraction procedure from the 3D coordinate is that each node connects with other nodes and itself, where we considered three kinds of edges: spatial, temporal and self connected edge. We explained the mathematical concept for set of edges E as follows:

The connection of two different nodes at the same time step known as a spatial edge which is defined by

The connection of two different nodes at the different time steps, known as the temporal edge, which is defined by .

The same node is connected with itself, known as a self-connected edge which is defined by

### Position Embedding

The recurrent network like GRUs and LSTM sequentially process the input, whereas our architecture is one kind of transformer that will not process the skeleton joint sequentially. We used positional embedding here to maintain the sequence of the joint information since there is no built-in notion of the sequence in the transfer. Each skeleton joint of the hand gesture is composed of a tensor for feeding the deep neural networks. For each node, there are no pre-defined structures or orders for showing the node's identity, and it's impossible to identify the corresponding node's hand gesture name. We need to provide unique markers or identifiers for every node to identify the gesture name of the corresponding node. We propose a spatial, temporal position encoding technique to generate the gesture information according to joint information. We use the sine and cosine functions by following [30,31,63,64] with various frequencies to encode the position number for each node as the encoding functions:

(4)

Here, represent the sin function position encoding for the odd index, represent the cos function position encoding for the odd index, the position encoding vector dimension is represented by i, and p denotes the position of each element. According to [63,64], the input hand skeleton contains space and time information, and one of the important strategies of the position embedders important strategies is to unify the spatial and temporal information and encode them sequentially. The spatial position embedding comprises the N vectors, where each individual vector consists of a hand joint. We applied spatial position encoding by joining all joints in a single frame by encoding sequentially. On the other hand, temporal position embedding is composed of individual vectors, and each vector represents the corresponding node's hand skeleton graph. We encoded them by encoding the same joints in different frames.

Lastly, we added the position information with the output of the NN1 network which is considered an initial feature of a specific node and fed into the proposed architecture after being embedded with the associated position vector. We added the feature vector with the embedding position using the following formula:

(5)

(6)

Here, Equation (5),(6) produces the final feature and for spatial-temporal and temporal-spatial for a specific node respectively. In the equation, represent the initial feature, represent output of temporal attention, represent the output of spatial attention. The hand joint of the t frame is represented here by and where the embedding dimension is the same as the input dimension.

### Spatial-Temporal Attention Module

The proposed approach consists of spatial-temporal, temporal-spatial attention and general deep neural network branches. Attention-based branches comprise the two-attention model with spatial embedding and two attention models with temporal embedding. In the first branch, the spatial attention block took the input from the output node of NN1 and updated them with the encoding spatial information with the spatial attention block; then, it is fed into temporal attention for updating with the temporal attention block and produced the spatial-temporal feature. In the same way, the 2nd branch produced the temporal-spatial feature by the reverse procedure of the 1st branch. In all cases, we applied a multi-head attention mechanism [21,30,31], which is visualized in Figure 5.

Consider, is the initial feature of a node of a hand skeleton, which is use as an input value of the attention layer.

There are multi-head in the attention mechanism and let -th attention head firstly apply the fully-connected layer for mapping query, key and value vectors with the input features. The mapping procedure was performed using the following formulas:

(7)

Here query, key and value nodes are represented by , , and respectively. The weight matrix of the fully connected layer for the m-th spatial or temporal attention model is denoted by ,,. The weights metrics of the spatial or temporal and self-connected edge is calculated in two stages. In the first stage, simultaneously calculates the dot-product between the query and the key vectors [21,30,31]. Using a SoftMax activation function normalize the output of the dot product.

Following formulas in the Equation (8) execute the above two steps:

, (8)

Where d represents the dimension of the key vectors and scaled dot products **between**  and nodes are represented by ; inner product operation is represented by ; attention operation is represented by , which is extracted effective information from v to node. In this stage, we can determine whether the attention will be considered spatial or temporal using masking operation by assigning a value of edges. We block the temporal domain information passing by assigning 0 weights for all temporal edges to consider spatial attention and vice versa.

Consequently, a weighted skeleton graph is produced by the spatial attention block by considering a hand joint for the same time frame, and the attention head calculates from the node using Equation (9).

(9)

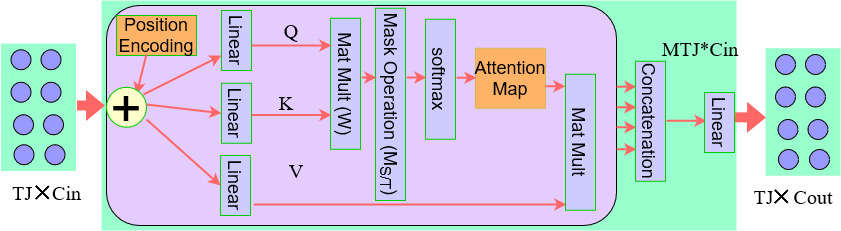


Figure 5: Attention Map Architecture with Masking Operation

Here, represent the output of the spatial attention for the node as we blocked all the temporal domain edges. Moreover, the main idea of spatial attention is to calculate the relationship between two nodes and information passing among the nodes within the same time steps. In addition, according to the learned edge weights, their aggregates and the received information.

Equation (9) repeated itself. M times for producing the multi-head attention of spatial or temporal domain, which is considered as multiple feature vectors. Finally, all the attention head outputs concatenate according to Equation (10) and make a single feature vector as which is considered the feature vector for the node and we considered the spatial attention feature .

(10)

Here, M is the total number of heads in multi-head attention in our case, 8. In the 1st branch, the spatial attention the model learns the weighted skeleton graphs and produces node features by encoding multiple types of structural information. The spatial attention feature is considered the input features for the temporal attention and employed the described multi-head attention procedure in the temporal domain and produced the spatial-temporal feature information. In the same way, in the 2nd branch the, temporal attention models learn weighted skeleton graphs and produce node features by encoding multiple types of structural information and then feeding it into the spatial attention and employed the described multi-head attention procedure in the temporal position embedding domain.

### Spatial-Temporal Mask Operation:

In the proposed architecture, we employed the attention block's spatial and temporal masking operation to cut down the computational cost. In spatial attention, the block mask operator assigns 1 for the spatial position and 0 for others. In the same way, the temporal attention block mask operator contains 1 for temporal value and 0 for other positions. After performing the mask operation, it reduces the data block's size and cuts down the system's computational cost. The concept of attention block is first to calculate three fully connected layers for query, key and values vectors. Then among the query vector and key vector, it calculated the dot product and was divided by the dimension of the key vector. Before the SoftMax activation function, we employed mask operation for both spatial and temporal domains to block unnecessary domains' edges by assigning 0. In Figure 6, we illustrated our masking operation [30,31].

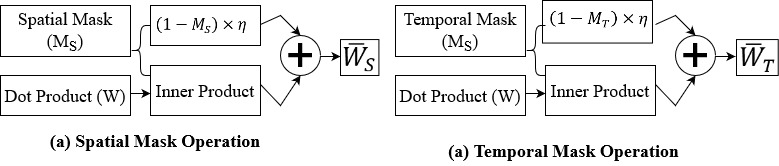


Figure 6: Spatial-Temporal Masking Operation

In the previous section, we discussed the attention where we computed a query matrix and key matrix. Each row of the matrix contained the query vector for each node, and each row of the contained the key for each node. Then we computed the edge weight matrix using scaled dot products by applying the following Equations:

(11)

Here, T represents the matrix transpose operation and represent the matrix multiplications. The edge weights can be contained a spatial edge or temporal edge depending on the setting value in each element of W. Here, in the first stage, we proposed spatial mask operations to set the value in W that contains the temporal edge to . Where the value of the is near zero and keeps other values unchanged. After applying the spatial mask, we got the output that contained spatial edge and contained the temporal edge. The following equations calculate the spatial edge and temporal edge:

(12)

Where the inner product or element-wise dot product is denoted by , and spatial mask operation is denoted by whose value depends on edge type, represent the weight matrix; if the edges are self-connected or spatial, then it's 1; otherwise, 0. At this work, we assign for the . The SoftMax activation is denoted by which normalizes the weights based on the spatial edges because the value of the eta is near zero. As a consequence, all temporal edge is set 0 at . Here, represent the spatial mask contains one if edge represents self-connected or spatial edge otherwise 0. The edge weight calculation formula in the spatial domain of Equation (8) is successfully implemented by Equations (11) and (12). However, masking output a matrix can be applicable for computing the node feature described in equation (12) based on the matrix multiplication with the value vectors matrix. In the same way, we employed temporal mask operation according to Equation (13) where we used instead for computing the weight matrix in the temporal domain.

(13)

According to the previous discussion, this matrix contains if it is temporal or self-connected edge; otherwise, 0. The main goal of the mask operation is to increase the efficiency of the system by reducing the computational complexity.

## General Deep Neural Network Branch

In our study, the general deep neural network branch is used as an alternative path to reach the output of NN2 to to concatenate with spatial-temporal and temporal-spatial features. In the NN1, we firstly employed a fully connected layer along with the relu function, then normalize with layer normalization and dropout layer were used to reduce the overfitting and produced the initial feature . In the NN2 taken output of NN1 as input with three dimensions where a fully connected layer produced 256 dimensions after applying layer normalization, then we employed an average pooling layer to produce an average vector, and finally, a padding layer was used for maintaining the output dimension general feature vector from the NN2. This branch effectively solves the missing data problems and converges problems for exploding gradient and vanishing gradient, which face difficulties in the other branch [65-66].

# Experimental Result

We evaluated the proposed model with three datasets here where the proposed model included spatial and temporal, two kinds of multi-head attention models with eight heads which took input from the output of the deep neural network and after average pooling, we employed a fully connected layer as the final layer for classification.

## Experimental configuration of training and testing

We implemented our architecture in the PyTorch platform in the NVIDIA GPU 8GB machines in the study. We randomly selected eight frames for each video as the input. First, we subtract every input frame sequence by the first frame palm position based on the previous work; then, we employed some data augmentation techniques by following previous work like shifting, scaling, time interpolation and adding noise. In the compiling section, we used Adam optimizer as an optimizer method with the .001 learning rate for training the model, where batch size was set to 32 and dropout rate was set to 0.1 and 0.2 [**67**].

## Experimental Setup and Implementation Protocols:

To evaluate the proposed model, we used three datasets, namely MSRA [52], DHG [13], and SHREC[14] dataset. Among them, DHG and SHREC both contain 2800 video sequences for 14 and 28 gestures, and 3D coordinates of 22 joints are collected for each frame. MSRA dataset is collected for 17 gestures and 500/600 frames for each of the gestures in 76500 frames. There were 9, 20 and 27 subjects for MSRA, DHG, and SHREC datasets, respectively.

We used all three datasets to evaluate our model with a cross validation procedure namely leave-one-out cross-validation (LOOCV). According the procedure, we sequentially selected n-1 subject information for training for each experiment and the remaining subject for testing. There are nine subjects in the MSRA dataset; keeping one subject dataset for testing we trained the model with remaining 9 subject dataset.

There are 20 subjects in the DHG dataset; we took one subject for testing and the remaining 19 for training. In the same way, among 27 subject datasets for the SHREC'17 dataset; we considered 26 subject datasets for training and the remaining one as testing dataset. The overall accuracy of all gestures is reported here.

## Experimental Result

## We showed the accuracy of three datasets our proposed MSTA method performed. In the first section, we demonstrated for MSRA; then, we demonstrated for DHG and the SHREC'17 dataset. In the SectionV-C1, we demonstrated for MSRA; then, we demonstrated for DHG and the SHREC’17 dataset in Section V-C2.

## Evaluation with MSRA Dataset

|  |  |
| --- | --- |
| Subject name | Accuracy (%) |
| Subject-1 | 100 |
| Subject-2 | 94.11 |
| Subject-3 | 94.11 |
| Subject-4 | 94.11 |
| Subject-5 | 100 |
| Subject-6 | 94.17 |
| Subject-7 | 88.24 |
| Subject-8 | 100 |
| Subject-9 | 82.35 |
| Average | **94.12** |

Table 2: Performance Accuracy for MSRA Dataset for 17 Gesture

## In the first stage, we evaluated our proposed system with the MSRA dataset where eight subject datasets used for the training and remaining subjects dataset for the evaluation. Table 2 shows the performance accuracy of the MSRA dataset where we reported nine individual subject performance accuracy and average accuracy among 9-subject as well. We got maximum accuracy of 100% for subject 1, subject five and subject eight and minimum accuracy got 82.35% accuracy at subject 9 and a 94.12% average accuracy for the nine subjects.

## Evaluation with DHG Dataset and SHREC'17 Dataset.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Subject name | DHG Dataset  Accuracy (%) | | SHRACE'17  Accuracy (%) | |
|  | 14 Class | 28 Class | 14 Class | 28 Class |
| Subject-1 | 85.00 | 87.02 | 99.16 | 95.00 |
| Subject-2 | 83.36 | 75.00 | 97.33 | 89.00 |
| Subject-3 | 96.77 | 85.71 | 97.49 | 92.49 |
| Subject-4 | 92.63 | 87.80 | 94.00 | 87.81 |
| Subject-5 | 91.26 | 91.38 | 96.05 | 94.09 |
| Subject-6 | 90.23 | 87.57 | 99.10 | 91.38 |
| Subject-7 | 91.10 | 85.20 | 98.40 | 95.74 |
| Subject-8 | 94.36 | 91.25 | 96.66 | 95.00 |
| Subject-9 | 93.57 | 94.15 | 99.76 | 95.05 |
| Subject-10 | 97.64 | 95.07 | 99.50 | 98.56 |

Table 3: Performance Accuracy for DHG and SHREC'17 Dataset for 14 and 28 Gestures

## Secondly, to evaluate the proposed model with another dataset namely DHG-14/28, we trained the model using 19 subject dataset and tested it using the remaining subject for each experiment. Accordingly, we repeated it 20 times and used different subjects for both DHG-14 and DHG-28. In the same way, for the SHREC'17-14/28 dataset we trained using 26 subjects’ information and tested it on the remaining ones. We repeated it 27 times accordingly for testing different subjects for both SHREC'17-14 and SHREC'17-28. Table 3 demonstrated the performance accuracy of the 10 subjects for both DHG and SHREC'17 datasets for 14 and 28 gestures. For the DHG dataset with 14 gestures, we got a 97.64% maximum accuracy at subject ten and minimum accuracy of 83.36% at subject 2. In the same way 28 classes of the DHG dataset, we got maximum accuracy of 95.05% at subject ten and minimum accuracy of 75.00% at subject 2. For the SHREC'17 dataset with 14 classes, we got 99.76% accuracy in subject 9, whereas minimum accuracy of 94.00% got in subject 4. For 28 classes of the SHREC'7 dataset, we got a maximum accuracy of 98.56% at subject 10, whereas the minimum accuracy was 87.81% at subject 4. The average accuracy for all 20 and 27 subjects is demonstrated in Table 5 and Table 6 for comparison.

## Comparison with State-of-the-Art Method

We compared our evaluation performance with the state-of-art model for the all datasets to prove the superiority of the proposed system. In the Section V-D1, V-D2,and V-D3 showed the comparison for MSRA, DHG and SHREC17 dataset, respectively.

**Comparison of MSRA Dataset:**

Our model produced good performance accuracy for the MSRA dataset by comparing the stat-of-the-art model shown in Table 4. The state-of-the-art model proposed by Ma et al. employed an enhanced neural al network, GREN and LSTM architecture to recognize hand gestures using a skeleton dataset based on the augmented neural network with one short learning memory [23]. The main goal of their idea is to improve performance accuracy, minimize prediction error, and remove unnecessary hyperparameter updating. Their model aims to design a network that can effectively combine and share the feature between dissimilar classes and experiment with their model in different ways. Based on the skeleton information, they employed an LSTM network where achieved 72.92% accuracy and achieved 79.17% accuracy with the green network. On the other side, our proposed study achieved 94.12% accuracy, which is more than 10.00% of the existing method.

|  |  |  |
| --- | --- | --- |
| Method Name | Class | Accuracy [%] |
| LSTM [23] | 17 | 72.92 |
| Green [23] | 17 | 79.17 |
| Proposed Attention | 17 | 94.12 |

Table 4: State of the Art comparison of the MSRA Dataset for 17 Gestures

**Comparison of DHG Dataset:**

In Table 5, proposed study compared with the various state-of-the-art method for DHG dataset for both 14 and 28 gestures. It demonstrated that the proposed study outperforms most state-of-the-art techniques and achieves comparable accuracy performance with DG-STA [30] and STA-GCN [8]. Although some existing methods used depth and skeleton both information, such as joint angles and HOG2 (JAHOG) [15] approaches, ASJT[38], SoCJ + HoHD + HoWR[13], NIUKF-LSTM[26], CNN+RNN[41], our study only relies on the only skeleton. Our method generated an average accuracy of 20 subjects at 92.00% for the 14-gesture, which is higher than the advanced algorithm. In the case of 28 gestures, it achieved 88.78% average accuracy for 20 subjects, which is also higher than the existing performance accuracy. JAHOG [15] applied joint similarity with [29] and achieved 83.35% and 76.53% accuracy sequentially for 14 and 28 gestures. In [16], they employed motion features augmented with RNN (MARNN), achieving 84.68% and 80.32% for 14 and 28 gestures sequentially. Smedt et al. also show satisfactory performance accuracy, but they showed a problem for incorrect joint locations during closed the hands [13]. They combined geometric features with the multi-level representation of the fisher vector that is ensured by the temporal pyramid to achieve the feature for SVM classifier. LSTM based technique although achieved better performance compared to the hand-crafted features such as CNN + LSTM [17], NIUKF-LSTM [**26**], Green [23], MFA-Net [27]. Ma et al. employed LSTM to handle noisy skeleton data by integrating a spatial type of kalman filter namely: nested interval unscented kalman filter (NIUKF), then achieved 8.92% and 80.44% accuracy for the 14 and 28 gestures of the DHG dataset sequentially [26].

|  |  |  |
| --- | --- | --- |
| **Methods** | **Accuracy (%)**  **(14 Gestures)** | **Accuracy (%)**  **(28 Gestures** |
| JAHOG [15] | 83.85% | 76.53% |
| GREN [23] | 82.29% | 82.03% |
| ASJT [38] | 82.50 | 80.11 |
| SoCJ + HoHD + HoWR [13] | 83.07 | 80.0 |
| MARNN[16] | 84.68 | 80.32 |
| CNN+RNN [41] | 85.46 | 74.19 |
| NIUKF-LSTM [26] | 84.92% | 80.44% |
| CNN+LSTM [17] | 85.46% | 74.19% |
| MFA-Net [27] | 85.75% | 81.04% |
| Res-TCN [28] | 86.90 | 83.60 |
| STA-Res-TCN [28] | 89.20 | 85.00 |
| Boulahia [29] | 90.48 | 80.48 |
| STA-GCN [8] | 91.20 | 87.1 |
| DG-STA [30] | 91.00 | 88.00 |
| Proposed Method | **92.00** | **88.78** |

Table 5: State of the Art comparison of the DHG Dataset

Nunez et al. produced accuracy by combining CNN and LSTM, where they focused on spatiotemporal-based skeleton features and achieved higher performance than hand-crafted features [17].

For recognizing hand-gesture using skeleton hand joint motion feature augmented network (MFA-Net) model is proposed by Chen et al. and achieved 85.75% and 81.10% for the DHG dataset for 14 and 28 gestures sequentially [27]. Another technique employed by Ma et al. based on the GREN and LSTM architecture to recognize hand gestures and achieved 82.29% and 82.03% for the DHGD skeleton dataset [23]. Res-TCN [28], STA-Res-TCN [28], STA-GCN [8] and DG-STA [28] are applied attention-based architecture for recognizing hand gestures based on skeleton information. How et al. employed a spatial-temporal attention-based neural network: STA-Res-TCN for extracting feature from different level of attention block for each time steps and achieved 89.20% and 85.00% for 14 and 28 gestures sequentially for the DHG skeleton hand gesture dataset [28]. Boulahia et al. extracted Hif3d for gesture classification and achieved 90.48% for 14 gestures and 80.48% for the 28 gestures of the DHG dataset [29]. Chen et al. employed the DG-STA approach to improve accuracy and reduce the computational cost for hand gesture recognition and achieved 91.00% and 88.00% accuracy for the DHG dataset [30]. Unlike existing work, our proposed architecture focuses on multiple branches for producing multiple feature vectors generated by the parallel architecture, which also preserves the properties of dynamic hand gestures properties. **Moreover,** replacing some branches of the proposed architecture can easily be compatible with the existing state-of-the-art system like DG-STA [30].

Moreover, our study's main focus is to fully explore the composition of prior work and future work. The table's contents have demonstrated that our proposed method's performance is higher than the existing method in this factor.

## Comparison of SHREC'17 Dataset.

The comparison Table 6 demonstrated that our model outperforms most of the state-of-the-art methods for the SHREC'17 dataset for both 14 and 28 gesture cases and comparable performance with DG-STA [30] and STA-GCN [8]. As shown in Table 4, our study achieved 97.01% for 14 and 92.78% accuracy for the 28 gestures, which is average for 27 subjects and outperformed all existing methods for both experiment settings. Specifically, MSTA improved the accuracy of 14 gestures by 3.40% and 4.40% for the 28 gestures once we compared them with the existing best-performance DG-STA [30] methods and more than 5.40% with more recent work by STA-GCN [8].

Although some existing methods used depth and skeleton, both information among them, histogram-based method based on depth sequence (HON4D) [32], shape analysis of motion trajectories on riemannian manifold (SMTRM) [33] for hand gesture classification, SoCJ + HoHD + HoWR [14], while our study only relies on an only skeleton. In the case of the SHREC17 dataset, MFA-Net produced 91.31% and 86.55% accuracy for 14 and 28 gestures sequentially [ 27]. Res-TCN, STA-Res-TCN [28], STA-GCN [8] and DG-STA [30] are applied attention-based architecture for recognizing hand gestures based on skeleton information. Among the attention based model, STA-Res-TCN achieved 93.60% and 90.70% accuracy for 14 and 28 gestures [28] where as DG-STA [30] approach to improve accuracy and reduce the computational cost of hand gesture recognition and achieved 94.40% and 90.00% accuracy for sequentially the 14 and 28 gestures.

Our proposed method mainly focuses on parallely producing multiple features from multiple branches of the parallel architecture, which preserves the properties of dynamic hand gestures. In addition, our study can be compatible with the existing attention-based method discarding some branches and modules [28, 29, 30]. Moreover, our study's primary focus is to explore the composition of prior work and future work. The table's contents have demonstrated that our proposed method's performance is higher than the existing method in this factor.

|  |  |  |
| --- | --- | --- |
| **Method** | **Accuracy (%)**  **(14 Gestures)** | **Accuracy (%)**  **(28 Gestures** |
| HON4D [32] | 78.53 | 74.03 |
| SMTRM [33] | 79.61 | 62.00 |
| SoCJ + HoHD + HoWR [14] | 88.24 | 81.90 |
| Res-C3D [41] | 89.52 | - |
| **MFA-Net [27]** | 91.31 | 86.55 |
| Res-TCN [28] | 91.10 | 87.3 |
| STA-Res-TCN [28] | 93.60 | 90.70 |
| STA-GCN [8] | 92.27 | 87.7 |
| DG-STA [30] | 94.4 | 90.00 |
| Proposed Method | 97.01 | 92.78 |

Table 6: State of the Art comparison of the SHREC'17 dataset

# Conclusion

We employed a attention based Multi-Branch Attention Based Graph and General Deep Learning approach for recognizing hand gestures based on the 3D hand skeleton data points in the study. Our method provided a multi-branch graph-based deep neural network and general deep neural network model with masking operation for learning spatial and temporal domain information and produced a potential feature vector for classification. We employed two branches of graph-based neural networks where the spatial-temporal branch took input from the output of the neural network NN1, and after encoding with spatial and temporal attention, it produced the spatial-temporal. In the same way 2nd branch produced a temporal-spatial feature by following the reverse sequence of the first branch. Which is concatenated with the output of the general deep neural network branch and applied average pooling layer. Finally, a fully connected layer is applied to learn node and edge weight for classification. Our proposed method achieved high performance for three datasets compared to the state-of-the-art model. In the table, we demonstrated the experimental result for three datasets and the effectiveness of our proposed architecture. In the future, we plan to collect 3D hand skeleton information from ourselves from more gestures to develop a sign language-based communication system.

APPENDIX A

The following acronyms are used in this paper

|  |  |
| --- | --- |
| Abbreviation | Description |
| NN1 | Deep Neural Network-1 |
| NN2 | Deep Neural Network-2 |
| MSRA | Microsoft Research Asia |
| DHG | Dynamic Hand Gesture |
| SHREC | A name of the Data Collection Contest. |
| RGB-D | Red, Green Blue with Depth |
| CNN, | Convolutional Neural Network |
| SVM | Support Vector Machine |
| HON4D | Histogram-based Method Based on Depth Sequence |
| Res-C3D | 3D Convolutional Neural Networks with Residual Architecture |
| MFA-Net | Motion Feature Augmented Network |
| DHGD | Dynamic Hand Gesture Depth |
| MANS | Memory Attention Networks |
| GCNN | Graph Convolutional Neural Network |
| DG-STA | Dynamic Graph based Spatial Temporal Attention |
| ASJT | Analysing Set-of-Joints Trajectories |
| NIUKF-LSTM | Nested Interval Unscented Kalman Filter LSTM |
| Abbreviation | Description |
| GCN | Graph Convolutional Network |
| RNN | Recurrent Neural Network |
| NLP | Natural Language Processing |
| OAK-D | OpenCV AI Kit with Depth |
| RNNG | Recurrent Neural Network Grammar |
| LSTM | Long Short-Term Memory |
| HOG | Histogram of Oriented Gradients |
| SoCJ | Shape of Connected Joints |
| HoHD | Histogram of Hand Directions |
| HoWR | Histogram of Wrist Rtations |
| UKF | Unscented Kalman Filter |
| GREEN | Gesture Recognition using an Enhanced Network |
| Hif3d | Handwriting-inspired Features |
| STA-Res-TCN | Spatial-Temporal Attention by combining with Residual Connection and Temporal Convolutional Neural Network |
| JAHOG | Joint angle HOG |
| STA- GCN | Spatial Temporal Attention with Graph Convolutional Network |
| MARNN | Motion Features Augmented with RNN |
| SMTRM | Shape Analysis of Motion Trajectories on Riemannian Manifold |
| SHREC2017 | 3D Shape RetrievalContest 2017 |

# References

1. **Abu Saleh Musa Miah**, Jungpil Shin, Md A.M. Hasan, and Md A. Rahim. "BenSignNet: Bengali Sign Language Alphabet Recognition Using Concatenated Segmentation and Convolutional Neural Network" *Applied Sciences* 12, no. 8: 3933. April 2022. [SCI indexed]
2. **Abu Saleh musa miah**, Md. Al Mehedi Hasan, Abdur Rahim, Jungpil SHIN, Yuichi Okuyama. “Rotation, Translation and Scale-Invariant Sign Word Recognition Using Efficient Segmentation and Deep Learning**”** Computer Systems Science and Engineering, Tecscience [SCI indexed]
3. Md Abdur Rahim, Abu Saleh Musa Miah, Jungpil Shin, "Hand gesture recognition based on optimal segmentation techniques in human-computer interaction", 3rd IEEE International Conference on Knowledge Innovation and Invention (IEEE ICKII 2020), K200134, Aug. 21-23, 2020, Kaohsiung, Taiwan.
4. Khan, M.A., Mittal, M., Goyal, L.M. *et al.* A deep survey on supervised learning based human detection and activity classification methods. *Multimed Tools Appl* **80**, 27867–27923 (2021). https://doi.org/10.1007/s11042-021-10811-5
5. G. Devineau, F. Moutarde, W. Xi and J. Yang, "Deep Learning for Hand Gesture Recognition on Skeletal Data," *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)*, 2018, pp. 106-113, doi: 10.1109/FG.2018.00025.
6. Siddharth S Rautaray and Anupam Agrawal. Vision based hand gesture recognition for human computer interaction: a survey. Artificial Intelligence Review, 43(1):1–54, 2015.
7. Johansson, G. Visual perception of biological motion and a model for its analysis. *Perception & Psychophysics* **14,** 201–211 (1973). <https://doi.org/10.3758/BF03212378>
8. Sijie Yan, Yuanjun Xiong, and Dahua Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. In Proceedings of the AAAI Conference on Artificial ntelligence, 2018.
9. Chenyang Si, Ya Jing, Wei Wang, Liang Wang, and Tieniu Tan. Skeleton-based action recognition with spatial reasoning and temporal stack learning. In Proceedings of the European Conference on Computer Vision (ECCV), pages 103–118, 2018.
10. Shi, L., Zhang, Y., Cheng, J., Lu, H.: Two-Stream Adaptive Graph Convolutional Networks for Skeleton-Based Action Recognition. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2019) 12026–12035
11. Oberweger, M.;Wohlhart, P.; Lepetit, V. Training a feedback loop for hand pose estimation. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 3316–3324. [CrossRef
12. Markus Oberweger and Vincent Lepetit. Deepprior++: Improving fast and accurate 3D hand pose estimation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 585–594, 2017
13. De Smedt, Q.; Wannous, H.; Vandeborre, J.P. Skeleton-based dynamic hand gesture recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 1–9
14. Q. De Smedt, H. Wannous, J.-P. Vandeborre, J. Guerry, B. Le Saux, and D. Filliat. Shrec’17 track: 3d hand gesture recognition using a depth and skeletal dataset. In *10th Eurographics Workshop on 3D Object Retrieval*, 2017.
15. Ohn-Bar, E.; Trivedi, M. Joint angles similarities and HOG2 for action recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Portland, OR, USA, 23–28 June 2013; pp. 465–470. [CrossRef]
16. Chen, X.; Guo, H.;Wang, G.; Zhang, L. Motion feature augmented recurrent neural network for skeleton-based dynamic hand gesture recognition. In Proceedings of the IEEE International Conference on Image Processing, Beijing, China, 17–20 September 2017; pp. 2881–2885. [CrossRef]
17. Nunez, J.C.; Cabido, R.; Pantrigo, J.J.; Montemayor, A.S.; Velez, J.F. Convolutional neural networks and long short-term memory for skeleton-based human activity and hand gesture recognition. Pattern Recognit. **2018**, 76, 80–94. [CrossRef]
18. Zhang, P., Lan, C., Xing, J., Zeng, W., Xue, J., Zheng, N.: View Adaptive Recurrent Neural Networks for High Performance Human Action Recognition From Skeleton Data. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2017) 2117–2126
19. Qiu, Z., Yao, T., Mei, T.: Learning Spatio-Temporal Representation with Pseudo3D Residual Networks. In: The IEEE International Conference on Computer Vision (ICCV). (2017) 5533–5541 9.
20. Y. Min, Y. Zhang, X. Chai and X. Chen, "," 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020, pp. 5760-5769, doi: 10.1109/CVPR42600.2020.00580. 59
21. Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In Advances in Neural Information Processing Systems (NIPS), pages 5998–6008, 2017 -18
22. Dai, Z., Yang, Z., Yang, Y., Cohen, W.W., Carbonell, J., Le, Q.V., Salakhutdinov, R.: Transformer-xl: Attentive language models beyond a fixed-length context. arXiv:1901.02860 (2019) 19
23. Ma, C.; Zhang, S.; Wang, A.; Qi, Y.; Chen, G. Skeleton-Based Dynamic Hand Gesture Recognition Using an Enhanced Network with One-Shot Learning. Appl. Sci. **2020**, 10, 3680. Green 20
24. Chenyang Si, Ya Jing, Wei Wang, Liang Wang, and Tieniu Tan. Skeleton-based action recognition with spatial reasoning and temporal stack learning. In Proceedings of the European Conference on Computer Vision (ECCV), pages 103–118, 2018 61
25. A. Bigalke and M. P. Heinrich, "Fusing Posture and Position Representations for Point Cloud-Based Hand Gesture Recognition," 2021 International Conference on 3D Vision (3DV), 2021, pp. 617-626, doi: 10.1109/3DV53792.2021.00071.
26. Ma, C.; Wang, A.; Chen, G.; Xu, C. Hand joints-based gesture recognition for noisy dataset using nested interval unscented Kalman filter with LSTM network. Visual Comput. **2018**, 34, 1053–1063. 22
27. Chen, X.; Wang, G.; Guo, H.; Zhang, C.; Wang, H.; Zhang, L. MFA-Net: Motion feature augmented network for dynamic hand gesture recognition from skeletal data. Sensors **2019**, 19, 239. [CrossRef] 23
28. Jingxuan Hou, Guijin Wang, Xinghao Chen, Jing-Hao Xue, Rui Zhu, and Huazhong Yang. Spatial-temporal attention Res-TCN for skeleton-based dynamic hand gesture recognition. In Proceedings of the European Conference on Computer Vision (ECCV), pages 273–286, 2018. 24
29. Boulahia, S.Y.; Anquetil, E.; Multon, F.; Kulpa, R. Dynamic hand gesture recognition based on 3D pattern assembled trajectories. In Proceedings of the 7th IEEE International Conference on Image Processing Theory, Tools and Applications (IPTA 2017), Montreal, QC, Canada, 28 November–1 December 2017. 25
30. Chen Y, Zhao L, Peng X, Yuan J, Metaxas DN. Construct dynamic graphs for hand gesture recognition via spatial-temporal attention. arXiv preprint arXiv:1907.08871. 2019 Jul 20. 26
31. Shi, L., Zhang, Y., Cheng, J., Lu, H. (2021). Decoupled Spatial-Temporal Attention Network for Skeleton-Based Action-Gesture Recognition. In: Ishikawa, H., Liu, CL., Pajdla, T., Shi, J. (eds) Computer Vision – ACCV 2020. ACCV 2020. Lecture Notes in Computer Science(), vol 12626. Springer, Cham. <https://doi.org/10.1007/978-3-030-69541-5_3> 21
32. Chunyu Xie, Ce Li, Baochang Zhang, Chen Chen, Jungong Han, Changqing Zou, and Jianzhuang Liu. Memory attention networks for skeleton-based action recognition. In IJCAI, 2018. 27
33. Kalpit Thakkar and P J Narayanan. Part-based graph convolutional network for action recognition. In BMVC, 2018. 28
34. Shan Lu, Dimitris Metaxas, Dimitris Samaras, and John Oliensis. Using multiple cues for hand tracking and model refinement. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), volume 2, pages 443–450, 2003.
35. Mohamed E. Hussein, Marwan Torki, Mohammad A. Gowayyed, and Motaz El-Saban. Human action recognition using a temporal hierarchy of covariance descriptors on 3d joint locations. In IJCAI, 2013. 35
36. Raviteja Vemulapalli, Felipe Arrate, and Rama Chellappa. Human action recognition by representing 3d skeletons as points in a lie group. In CVPR, 2014.
37. J. Wang, Z. Liu, Y. Wu, and J. Yuan. Mining actionlet ensemble for action recognition with depth cameras. In CVPR, 2012.
38. De Smedt, Q.; Wannous, H.; Vandeborre, J.P. 3D Hand Gesture Recognition by Analysing Set-of-Joints Trajectories. In Proceedings of the International Conference on Pattern Recognition (ICPR)/UHA3DS 2016 Workshop, Cancun, Mexico, 4 December 2016. 38
39. Li, S., Li, W., Cook, C., Zhu, C., Gao, Y.: Independently recurrent neural network (indrnn): Building A longer and deeper RNN. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2018) 5457–5466 14
40. Si, C.; Chen,W.;Wang,W.;Wang, L.; Tan, T. An attention enhanced graph convolutional lstm network for skeleton-based action recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 1227–1236.
41. Lin, C.; Lin, X.; Xie, Y.; Liang, Y. Abnormal gesture recognition based on multi-model fusion strategy. Mach. Vision Appl. **2019**, 30, 889–900. [CrossRef] 41
42. Lai, K.; Yanushkevich, S.N. CNN + RNN Depth and Skeleton based Dynamic Hand Gesture Recognition.In Proceedings of the 24th International Conference on Pattern Recognition (ICPR), Beijing, China, 20–24 August 2018. 42
43. Yu Tian, Xi Peng, Long Zhao, Shaoting Zhang, and Dimitris N Metaxas. CR-GAN: Learning complete representations for multi-view generation. In Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI), pages 942–948, 2018. 27
44. Liu, J.; Shahroudy, A.; Xu, D.;Wang, G. Spatio-temporal lstm with trust gates for 3d human action recognition. Proceedings of 14th European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016; pp. 816–833. [CrossRef]
45. Santoro, A., Raposo, D., Barrett, D.G., Malinowski, M., Pascanu, R., Battaglia, P., Lillicrap, T.: A simple neural network module for relational reasoning. In Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R., eds.: Advances in Neural Information Processing Systems. (2017) 4974–4983 18.
46. Hu, H., Gu, J., Zhang, Z., Dai, J., Wei, Y.: Relation Networks for Object Detection. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2018) 19.
47. Fu, J., Liu, J., Tian, H., Li, Y., Bao, Y., Fang, Z., Lu, H.: Dual attention network for scene segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2019) 3146–315 31
48. Zhengyuan Yang, Yuncheng Li, Jianchao Yang, , and Jiebo Luo. Action recognition with spatio-temporal visual attention on skeleton image sequences. IEEE Transactions on Circuits and Systems for Video Technology, 2018 43
49. Yong Du, Yun Fu, , and Liang Wang. Skeleton based action recognition with convolutional neural network. In ACPR, 2015.
50. Qiuhong Ke, Mohammed Bennamo memory attention networks un, Senjian An, Ferdous Sohel, and Farid Boussaid. A new representation of skeleton sequences for 3d action recognition. In CVPR, 2017.
51. Chao Li, Qiaoyong Zhong, Di Xie, and Shiliang Pu. Cooccurrence feature learning from skeleton data for action recognition and detection with hierarchical aggregation. In IJCAI, 2018
52. Fabien Baradel, Christian Wolf, and Julien Mille. Human action recognition: Pose-based attention draws focus to hands. In ICCV Workshop, 2017.
53. Sijie Song, Cuiling Lan, Junliang Xing, Wenjun Zeng, and Jiaying Liu. An end-to-end spatio-temporal attention model for human action recognition from skeleton data. In AAAI, 2017.
54. Chenyang Si, Ya Jing, Wei Wang, Liang Wang, and Tieniu Tan. Skeleton-based action recognition with spatial reasoning and temporal stack learning. In ECCV, 2018
55. Sun, X.; Wei, Y.; Liang, S.; Tang, X.; Sun, J. Cascaded hand pose regression. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 824–832. [CrossRef]
56. [A]. L. Seidenari, V. Varano, S. Berretti, A. Del Bimbo, and P. Pala, “Recognizing actions from depth cameras as weakly aligned multipart bag-of-poses,” in Proc. Comput. Vis. Pattern Recognit. Workshop (CVPRW) Human Activity Underst. 3D Data, Portland, OR, USA, Jun. 2013, pp. 479–485.
57. [B]. L. Xia, C.-C. Chen, and J. K. Aggarwal, “View invariant human action recognition using histograms of 3D joints,” in Proc. Workshop Human Activity Underst. 3D Data, Providence, RI, USA, Jun. 2012, pp. 20–27.
58. [C]. C. Ellis, S. Z. Masood, M. F. Tappen, J. J. La Viola, Jr., and R. Sukthankar, “Exploring the trade-off between accuracy and observational latency in action recognition,” Int. J.Compute. Vis., vol. 101, no. 3, pp. 420–436, 2013.
59. [D]Shahroudy, A., Liu, J., Ng, T.T., Wang, G.: NTU RGB+D: A Large Scale Dataset for 3D Human Activity Analysis. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2016) 1010–1019
60. [E]. Tompson, M. Stein, Y. Lecun, and K. Perlin. Real-time continuous pose recovery of human hands using convolu- tional networks. ACM Transactions on Graphics (TOG),2014.
61. [F] D. Tang, H. J. Chang, A. Tejani, and T.-K. Kim. Latent re- gression forest: Structured estimation of 3D articulated handposture. In CVPR, 2014 58
62. Pavlo Molchanov, Xiaodong Yang, Shalini Gupta, Kihwan Kim, Stephen Tyree, and Jan Kautz. Online detection and classification of dynamic hand gestures with recurrent 3d  
    convolutional neural network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 4207–4215, 2016. 60
63. Chen, X.; Wang, G.; Guo, H.; Zhang, C. Pose guided structured region ensemble network for cascaded hand pose estimation. Neurocomputing **2019**, 395, 138–149. [CrossRef] 62
64. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is All you Need. In Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R., eds.: Advances in Neural Information Processing Systems. (2017) 6000–6010 63
65. Mahmud, Hasan & Morshed, Mashrur & Hasan, Md Kamrul. (2021). A deep-learning--based multimodal depth-aware dynamic hand gesture recognition system. 64
66. Petar Velickovi ˇ c, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, ´ and Yoshua Bengio. Graph attention networks. arXiv preprint arXiv:1710.10903, 2017 65
67. Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980, 2014