applied sciences

MDPI

Article

Skeleton-Based Dynamic Hand Gesture Recognition Using an Enhanced Network with One-Shot Learning

Chunyong Ma [1,2,*], Shengsheng Zhang [1], Anni Wang [1], Yongyang Qi [1] and Ge Chen [1,2]

[1] College of Information Science and Engineering, Ocean University of China, Qingdao 266100, China; zhangshengsheng@stu.ouc.edu.cn (S.Z.); wanganni@stu.ouc.edu.cn (A.W.); qiyongyang@ouc.edu.cn (Y.Q.); gechen@ouc.edu.cn (G.C.)

[2] Laboratory for Regional Oceanography and Numerical Modeling, Qingdao National Laboratory for Marine Science and Technology, Qingdao 266200, China

* Correspondence: chunyongma@ouc.edu.cn

Received: 12 April 2020; Accepted: 20 May 2020; Published: 24 May 2020

Abstract: Dynamic hand gesture recognition based on one-shot learning requires full assimilation of the motion features from a few annotated data. However, how to effectively extract the spatio-temporal features of the hand gestures remains a challenging issue. This paper proposes a skeleton-based dynamic hand gesture recognition using an enhanced network (GREN) based on one-shot learning by improving the memory-augmented neural network, which can rapidly assimilate the motion features of dynamic hand gestures. Besides, the network effectively combines and stores the shared features between dissimilar classes, which lowers the prediction error caused by the unnecessary hyper-parameters updating, and improves the recognition accuracy with the increase of categories. In this paper, the public dynamic hand gesture database (DHGD) is used for the experimental comparison of the state-of-the-art performance of the GREN network, and although only 30% of the dataset was used for training, the accuracy of skeleton-based dynamic hand gesture recognition reached 82.29% based on one-shot learning. Experiments with the Microsoft Research Asia (MSRA) hand gesture dataset verified the robustness of the GREN network. The experimental results demonstrate that the GREN network is feasible for skeleton-based dynamic hand gesture recognition based on one-shot learning.

Keywords: one-shot learning; gesture recognition; GREN; skeleton-based

1. Introduction

With the rapid development of Kinect, Leap Motion, and other sensors in recent years, hand motion capture is getting much more efficient. By estimating the posture of the hand gesture, the position information of each joint can be detected from video or image sequences. Recent research [1–5] has tried various ways for dynamic hand gesture recognition based on 3D skeleton data characterized as strong correlations, temporal continuity, and co-occurrence relationships. Besides, the skeleton-based algorithm has fewer parameters, which is easier to calculate and more suitable for analyzing dynamic hand gestures. However, it is still challenging because hands are non-rigid objects, which can express a variety of different semantics [6]. With the gesture recognition technology being applied in more fields such as gaming and industry training, it is often necessary to make different customized annotation samples in large sizes. However, it is worth noting that the existing hand gesture database could not meet the needs of gesture interaction in various fields. The cost of large-scale gesture sample extraction artificially in each field is so high that it would limit the application of gesture recognition [7,8]. Meanwhile, the traditional gradient-based networks also require extensive iterative training to complete the model optimization. When encountering the new data, the models need to

relearn their hyper-parameters to adequately incorporate the new information without catastrophic interference [9], which is inefficient. The existing networks fail to complete the optimization of the model with small size training samples, while one-shot learning could infer results as expected [10]. Therefore, the method of one-shot learning can be used to solve the problem that the model could not be optimized by the insufficient samples of skeleton-based dynamic hand gestures.

However, if the current algorithm of "one-shot learning" is directly applied to the hand gesture recognition, there will be three gradient-based optimization problems. Firstly, due to the small amount of data, many advanced and mature algorithms, such as Momentum [11] and Adagrad [12], cannot be optimized in limited iterations; especially when encountering non-convex problems, many hyper-parameters cannot achieve convergence. Secondly, for different tasks, the parameters of the network need to be initialized randomly. If the amount of data is too small, the final model cannot achieve convergence. This can be alleviated by conducting transferring learning methods, such as fine-tuning [13,14]. Finally, for the traditional neural network, its memory storage is limited. Additionally, the process of learning a new set of patterns will suddenly and completely erase a network's knowledge of what it had already learned, which is referred to as catastrophic interference [15]. Therefore, we need to find a memory module that can be used for large-scale storage and can also be accessed for relevant information. The large capacity enhanced memory neural networks, such as a neural Turing machine (NTM) [16], provides a feasible method for one-shot learning combined with hand gesture recognition. The NTM provides the capability to quickly encode and retrieve new information by limiting the changes in the output of the network before and after the network update [15,17]. In addition, it can also eliminate gradient-based optimization problems. On this basis, Santoro [9] introduced a new and pure content-based method for accessing an external memory, which is different from previous methods, additionally using a memory location-based focusing mechanism. The method can rapidly bind never-before-seen information to the external memory after a single presentation and combines the gradient descent to slowly learn an abstract method for obtaining useful representations of raw data. As a result, it can accurately identify the categories of data that have occurred only once.

This paper focuses on the architecture of enhanced neural networks based on skeleton-based algorithms and one-shot learning. Based on the memory-augmented neural network (MANN) [9], we propose skeleton-based dynamic hand gesture recognition using an enhanced network (GREN). The long short-term memory (LSTM) network is selected as the controller of the GREN network to enhance the recognition and memory ability of the network. Compared with the MANN network, which was originally applied to image recognition, the proposed GREN network classifies hand gestures by identifying skeletal sequences. Through the recognition of the GREN network, we conduct experiments on a dynamic hand gesture dataset (DHGD) [18] to show the effectiveness of our method. Then, we implement our method on the Microsoft Research Asia (MSRA) hand gesture dataset [19] to verify its contributions.

The rest of this paper is organized as follows:

- Section 2 details the related work of skeleton-based dynamic hand gesture recognition and one-shot learning.
- The GREN network is introduced in Section 3.
- The experiments of skeleton-based dynamic hand gesture recognition are explained in detail in Section 4.
- In Section 5, results and discussion are presented.
- The conclusions are given in Section 6.

2. Related Work

2.1. Skeleton-Based Dynamic Hand Gesture Recognition

Much research has been focused on skeleton-based dynamic hand gesture recognition [20–29]. Chen X. et al. [30] proposed a skeleton-based dynamic hand gesture recognition algorithm that has also been suggested to surpass depth-based methods in the aspect of performance. Chin-Shyurng et al. [31] created a skeleton-based model by capturing the palm position, and the dynamic time-warping algorithm was applied to the recognition of disparate conducting gestures at various conducting speeds, which achieves real-time dynamic musical conducting gesture recognition. Ding, Ing-Jr et al. [32] designed an adaptive hidden Markov model (HMM)-based gesture recognition method with user adaptation (UA) to simplify large-scale video processing to realize the natural user interface (NUI) of a humanoid robot device. Similarly, Kumar, Pradeep et al. [33] used the HMM to identify occluded gestures in line with a robust position invariant sign language recognition (SLR) framework.

Additionally, some studies have employed deep learning methods to conduct skeleton-based dynamic hand gesture recognition. Mazhar, Osama et al. [34] proposed that humans need neither to wear any specific clothing (motion capture clothes or inertial sensors) nor to carry a special remote control or learn complex teaching instructions in gesture recognition. As a result, they developed a real-time, robust, and background-independent gesture detection module in the light of convolutional neural network (CNN) transmission learning. Chen, XH et al. [29] exploited motion features of traits and global movements to augment features of recurrent neural networks (RNNs) for gesture recognition and improve the classification performance. Lin, C et al. [35] proposed a novel refined fused model in combination with the masked Res-C3D network and skeleton LSTM for abnormal gesture recognition in RGB-D videos, which learns discriminative representations of gesture sequences in particular abnormal gesture samples by fusing multiple characteristics from different models. Based on a combination of a CNN network and an LSTM network, Nunez, JC et al. [36] proposed a deep learning-based approach for temporal 3D pose recognition problems, and the proposed network architecture does not need to be adapted to the type of activity or the gesture to be recognized, as well as the geometry of the 3D sequence data as input. So far, there is no available deep learning network that can be directly used for skeleton-based dynamic hand gesture recognition based on small size samples.

2.2. One-Shot Learning

The implementations of one-shot learning can be divided into statistics-based, weight-based matching, and meta-learning. For the statistics-based, Lake [37] adopted the Bayesian framework realized one-shot learning of handwritten character pictures based on the statistical point of view and the way humans learn things, triggering the new wave of one-shot learning.

Besides the above statistics, there are also many methods on the basis of weighted matching for one-shot learning, which performs certain criteria modeling on known samples and then determines the class according to the distance of samples. The most typical method is the k-nearest neighbor (KNN), which is a nonparametric estimation method that can directly employ distance to determine the category without prior training. Another method is to learn an end-to-end nearest neighbor classifier, which can not only quickly learn new samples but also have a great generalization of known samples. Snell et al. [38] carried out classification by calculating the distance from prototype representations of each class, which turns into the nearest neighbor classification in the metric space. While Koch et al. [39] performed efficacious feature extraction on new samples by limiting input methods, then used supervised metric learning based on twin networks to train and finally reused features extracted by that network for small or no sample learning. Similarly, Oriol Vinyals et al. [40] also utilized metric learning based on deep neuro features, which uses external memory to enhance the neural network that maps a small labeled support set and an unlabeled example to its label, obviating the need for fine-tuning to adapt to new class types.

Meta-learning, also known as "learning to learn", aims to train a model on a variety of learning tasks, such that it can solve new learning tasks using only a small number of training samples [41]. A neural network with memory can implement meta-learning, but its memory storage is limited. A large number of new features may exceed the memory storage capacity so that the network cannot learn new tasks. The NTM network can solve this problem, as it is capable of both long-term storage via slow updates of its weights and short-term storage via its external memory module [16]. Based on the NTM network, Santoro et al. [9] introduced a memory access module that emphasizes accurate encoding of relevant (recent) information and pure content-based retrieval to implement meta-learning. Besides, Ravi et al. [42] proposed an LSTM-based meta-learner model, whose parameterization allows it to learn appropriate parameter updates specifically for the scenario where a set amount of updates will be made, while also learning a general initialization of another learner (classifier) network that allows for quick convergence of training.

In general, the current one-shot learning-based methods are in a booming period. However, there is still no appropriate method for one-shot learning with skeleton-based hand gesture recognition. Therefore, this paper will study the current advanced achievements and propose a suitable algorithm to realize hand gesture recognition in line with one-shot learning.

3. Dynamic Hand Gesture Recognition with the GREN Network

By improving a MANN network, this paper implements the GREN network based on one-shot learning, which is a variant of the NTM network from Santoro et al. [9]. Compared with the MANN network originally applied to image recognition, the proposed GREN network classifies hand gestures by recognizing skeletal sequences. The structure of the GREN network is shown in Figure 1.

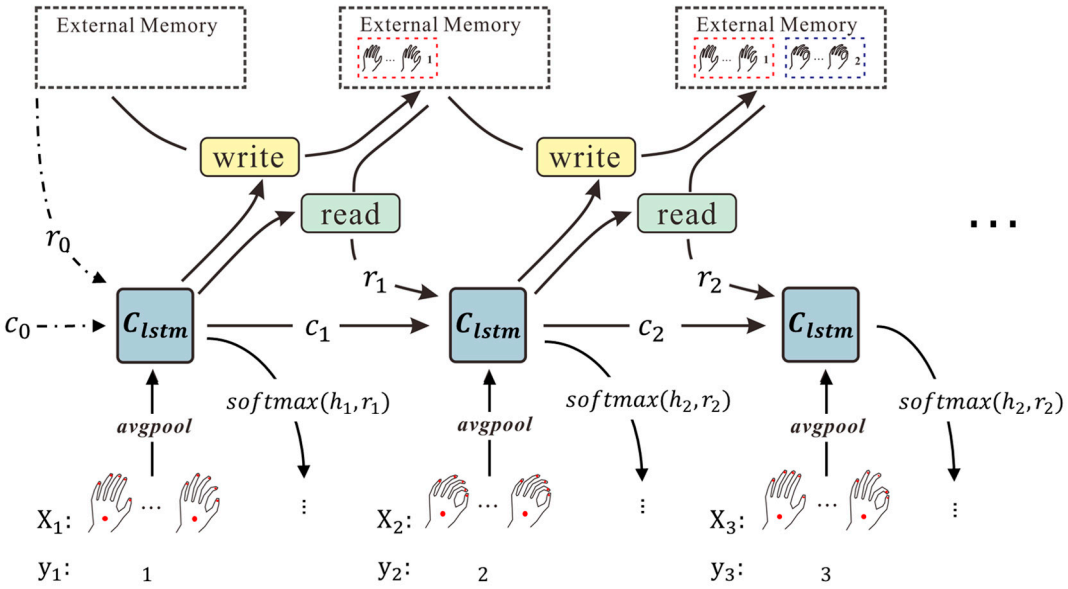

Figure 1. The structure of the GREN network. For the current time-step t, it takes the hand joint coordinate sequence X_t and the corresponding sample-class y_t as input and outputs the categorical distribution of prediction by a softmax layer. The controller, neuron C_{lstm}, generates h_t and c_t, which are the hidden state and the cell state of the LSTM used for the next time-step. A memory, r_t, is retrieved by the read heads from the external memory.

The GREN network consists of three components: a controller, read and write heads, and an external memory. The controller, neuron C_{lstm}, employed in our model is an LSTM network, which receives the current input and controls the read- and write-heads to interact with the external memory, respectively. Memory encoding and retrieval in an external memory are rapid, with vector representations being placed into or taken out of memory potentially every time-step [16], which

makes it a perfect candidate for one-shot prediction. Additionally, it can be stored either for long-term storage by slowly updating the weights or for short-term storage by an external memory. Thus, when the model learns the type of representation of a gesture sequence, it will be placed into memory, and later these representations will be used to make predictions of data that it has only seen once. Besides, according to the difference of classification methods between the input of images and sequences, the average pooling layer (avgpool) is introduced to further focus on characteristics of sequence and improve the calculation efficiency in the network. For one-shot learning, the output distribution is categorical, which is implemented as a softmax function.

At the beginning, the initialized state of the GREN network is represented by *init_state*. The external memory is initialized, which does not store any data representations. Also, the memory r_0 retrieved from the external memory is empty. In addition, the cell state of the initialized controller, neuron C_{lstm}, is represented by c_0. Given the input sequence X_t, the controller receives the memory r_{t-1} and cell state c_{t-1} provided by the previous state *prev_state*, then produces a query key vector k_t used to retrieve a particular memory. When encountering sequences of the already-seen class, the particular memory vector row could be retrieved by read heads, which is addressed using the cosine similarity measure:

$$\mathrm{K}(k_t, M_t(i)) = \frac{k_t \cdot M_t(i)}{||k_t|| \cdot ||M_t(i)||} \tag{1}$$

where M_t is the memory matrix at time-step t and $M_t(i)$ is the i^{th} row in this matrix. The row of $M_t(i)$ serve as memory "slots", with the row vectors themselves constituting individual memories.

After then, a read-weight vector w_t^r is produced by these similarity measures according to the softmax function:

$$w_t^r(i) \leftarrow \frac{\exp(\mathrm{K}(k_t, M_t(i)))}{\sum_j \exp(\mathrm{K}(k_t, M_t(j)))} \tag{2}$$

where the read heads can amplify or attenuate the precision of the focus by the read weights.

Those read weights w_t^r and corresponding memory $M_t(i)$ are used to retrieve the memory r_t:

$$r_t \leftarrow \sum_i w_t^r(i) \cdot M_t(i) \tag{3}$$

where the memory r_t is used by the controller as both an input to a classifier, namely, a softmax layer for class prediction and as an additional input for the next input sequence.

To achieve the combined learning in disparate classes and implement the one-shot learning, the least recently used access module (LRUA) proposed by Adam Santoro [9] is adopted, which is a pure content-based memory write head that writes memories to either the least used memory location or the most recently used one, and focusing on the accurate encoding of the most relevant information. In terms of a new sequence, it is written to a rarely-used location with the recently encoded information preserved or to the last used location, which can be used for updating with newer or possibly more relevant information:

$$w_t^u \leftarrow \gamma \cdot w_{t-1}^u + w_t^r + w_t^w \tag{4}$$

$$w_t^{lu}(i) = 1 \; if \; w_t^u \le m\left(w_t^u, n\right) \; else \; 0 \tag{5}$$

$$w_t^w \leftarrow \sigma(\alpha) \cdot w_{t-1}^r + (1 - \sigma(\alpha)) \cdot w_{t-1}^{lu} \tag{6}$$

$$M_t(i) \leftarrow M_{t-1}(i) + w_t^w(i) \cdot k_t \cdot \forall i \tag{7}$$

where w_t^u is the usage weight updated at each time-step to keep track of locations most recently read from or written to; γ is the decay parameter; w_t^{lu} is the least-used weight computed using w_t^u for a given time-step; the notation $m(v, n)$ is introduced to denote the n^{th} smallest element of the vector v; n is set to equal the number of the writer to memory; w_t^w is the written weight computed by the sigmoid function $\sigma(.)$, which combines the previous read weights w_{t-1}^r and previous least-used weights w_{t-1}^{lu}; α

is a dynamic scalar gate parameter to interpolate between weights. Before writing to memory, the least used memory location is computed from w^u_{t-1} and set it to zero, then the memory M_t is written by the computed vector of written weights w^w_t. Thus, $M_t(i)$ can be written into the zeroed memory location or the previously used memory location; if it is the latter, then w^{lu}_t will simply get erased.

With the above analysis, we propose the following GREN algorithm, as shown in Algorithm 1.

Algorithm 1: GREN

Input: Given N samples $\{X_1, X_2, \ldots, X_N\}$ belonging to C classes with
Sample-classes $y_t \in Y = \{1, \ldots, C\}$, $for\ t = 1, \ldots, N$;
Output: A softmax layer for class prediction;

Initialization:
$prev\_state \leftarrow init\_state(N)\{$
  $c_0 \leftarrow C_{lstm}(N)$;
  $r_0 \leftarrow 0_{N \times (head\_num * memory\_size)}$;
  $w^r_0 \leftarrow one\_hot\_weigh\_vector(N, head\_num, memory\_slots)$;
  $w^u_0 \leftarrow one\_hot\_weigh\_vector(N, memory\_slots)$;
  $M_0 \leftarrow \varepsilon_{N \times memory\_slots \times memory\_size}$;
  $return\ \{c_0, r_0, w^r_0, w^u_0, M_0\}$;
$\}$;
$o = [\,]$;
for $t \leftarrow 1$ **to** N **do**
  $h_t, c_t \leftarrow C_{lstm}((X_t, y_t), prev\_state)$;
  for $i \leftarrow 0$ **to** $X_t.length$ **do**
    $output, curr\_state \leftarrow gren((X_t(i), x\_lable_t(i)), prev\_state)\{$
      Memory Retrieval:
        $K(k_t, M_t(i)) \leftarrow cosine\_similarity(k_t, M_t(i))$;
        $w^r_t(i) \leftarrow softmax(K(k_t, M_t(i)))$;
        $r_t += w^r_t(i) \cdot M_t(i)$;
      Memory Encoding (LRUA):
        $w^u_t \leftarrow \gamma \cdot w^u_{t-1} + w^r_t + w^w_t$;
        if $w^u_t \le m\left(w^u_t, n\right)$ **then** $w^{lu}_t(i) = 1$ **else** $w^{lu}_t(i) = 0$;
        $w^w_t \leftarrow sigmoid(\alpha) \cdot w^r_{t-1} + (1 - sigmoid(\alpha)) \cdot w^{lu}_{t-1}$;
        $M_t(i) \leftarrow M_{t-1}(i) + w^w_t(i) \cdot k_t$;
      $return\ \{h_t, r_t\}, \left\{c_t, r_t, w^r_t, w^u_t, M_t\right\}$;
    $\}$;
    $prev\_state = curr\_state$;
    if $i == 0$ **then**
      $o2o\_w \leftarrow (output.length, M_{class}), rand\_unif\_init(minv, maxv)$;
      $o2o\_b \leftarrow (M_{class}), rand\_unif\_init(minv, maxv)$;
    end if;
    $output = output \cdot o2o\_w + o2o\_b$;
    $output = softmax(output)$;
    $o.append(output)$;
  end
  $learning\_loss = -cross\_entropy\_cost(y_t, o)$;
  $optimizer = AdamOptimizer(learning\_rate)$;
  $train\_op = optimizer.minimize(learning\_loss)$;
end

In the algorithm, the $one\_hot\_weigh\_vector(a, b, c)$ function generates a tensor of shape $a \times b \times c$ with $[:, :, 0]$ set to *one* (or $[:, 0]$, if the $one\_hot\_weigh\_vector(a, b)$ function generates a tensor of shape $a \times b$); $\{(a, b), rand\_unif\_init(minv, maxv)\}$ generates a tensor of shape $a \times b$ (or

$\{(a), rand\_unif\_init(minv, maxv)\}$ generates a tensor of shape $a \times 1$) with a uniform distribution, and the value of all elements is set between *minv* and *maxv*.

In general, for the current time-step t, the sample data X_t and the corresponding sample-class y_t will be received by the controller C_{lstm}. The current state of the GREN network *curr_state* is used by the controller as an additional input for the next time-step. According to each sequence of the sample, the GREN algorithm randomly generates the class label x_label_t. If the sample date X_t comes from a never-before-seen class, it will be bound to the appropriate sample-class y_t and stored by the write heads in the external memory, which is presented in the subsequent time-step (see Figure 1). Later, once a sample from an already-seen class is presented, the controller will retrieve the bound sample-class information by the read heads from the external memory for class prediction. A softmax layer, $softmax(\cdot)$, is selected to output the standardized probability distribution of the model prediction, and combined with the cross-entropy cost function, $cross\_entropy\_cost(\cdot)$, to measure the loss between the predicted value and correct class label. Then, the adaptive moment estimation (Adam) [43], $AdamOptimizer(\cdot)$, is adopted to minimize the loss, and the back-propagated error signal from the current prediction updates those previous weights, which is followed by the updating of the external memory. Those processes would be repeated until the model converges.

4. Experiments

In this section, two hand gesture datasets named dynamic hand gesture database (DHGD) and MSRA are used for the experiments. Details about the experimental setup of the GREN network are introduced in the later part of this section.

4.1. Datasets

4.1.1. DHGD Hand Gesture Dataset

The public DHGD hand gesture dataset [18] contains sequences for 14 right-hand gestures performed in two ways: using one finger and the whole hand. Each class of gestures is performed 1 to 10 times by 28 participants in both of the above two ways, resulting in 2800 sequences, and the length of the gestures varies from 20 to 50 frames. Each frame contains the coordinates of the 22 joints in the 2D depth image space and 3D world space, and those joints are shown in Figure 2.

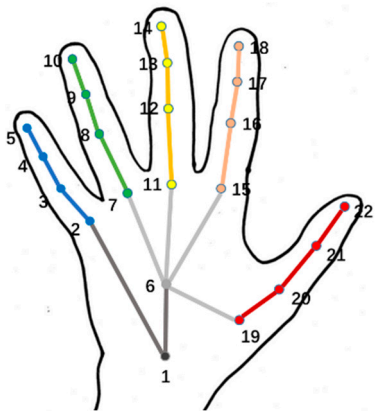

Figure 2. Twenty-two joints of a right-hand skeleton.

Some gestures (such as swipe and shake), which are defined by the movement of the hand, called the coarse gesture, while others are defined by the shape of the gesture, called the fine gesture. Table 1 shows the different classes of gestures in DHGD:

Table 1. List of 14 gestures in the dynamic hand gesture database (DHGD).

Name of the Gesture	Type of the Gesture	Type of the Gesture
1	Grab	Fine
2	Tap	Coarse
3	Expand	Fine
4	Pinch	Fine
5	Rotation Clockwise	Fine
6	Rotation Counter Clockwise	Fine
7	Swipe Right	Coarse
8	Swipe Left	Coarse
9	Swipe Up	Coarse
10	Swipe Down	Coarse
11	Swipe X	Coarse
12	Swipe +	Coarse
13	Swipe V	Coarse
14	Shake	Coarse

4.1.2. MSRA Hand Gesture Dataset

The public MSRA [19] hand gesture dataset, which contains skeleton-based sequence data of 17 right-hand gestures performed by 28 participants, is chosen to verify the robustness of the GREN network. The 17 right-hand gestures are manually chosen and are mostly from American Sign Language, to span the space of finger articulation as much as possible. Additionally, the length of each gesture varies from 490 to 500 frames. Each of these frames contains the coordinates of the 21 joints in the 2D depth image space and 3D world space, and those joints are shown in Figure 3.

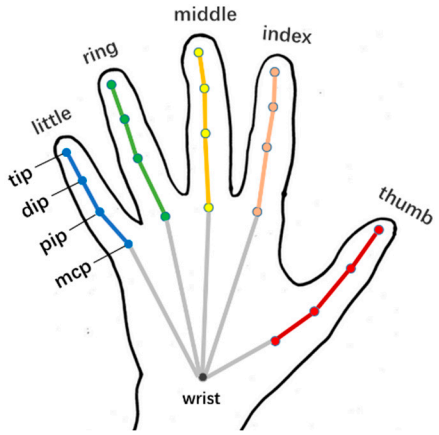

Figure 3. Twenty-one joints of a right-hand skeleton.

4.2. Experimental Setup

4.2.1. Data Pre-Process

The skeleton-based hand gesture datasets should be preprocessed as the input of our network. The whole framework of the data preprocessing is shown in Figure 4, in which the k^{th} class gesture is processed by our method as an example.

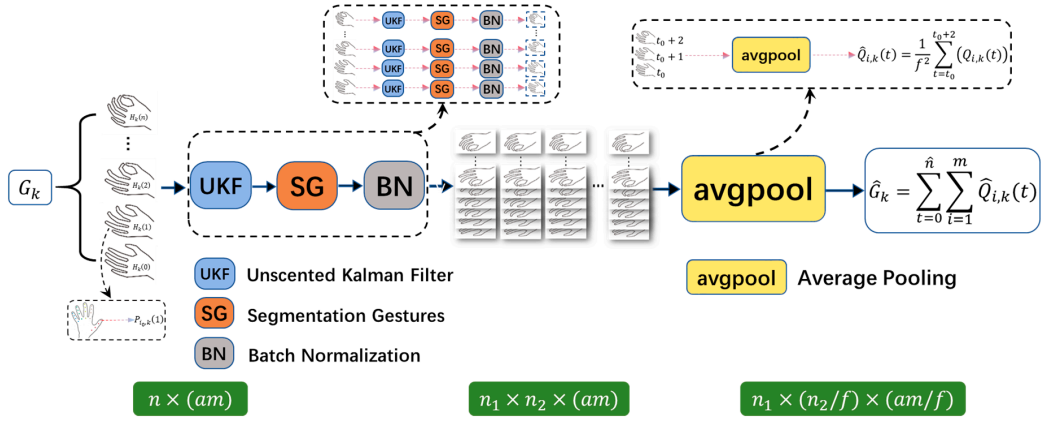

Figure 4. Framework of the data preprocessing.

First of all, the nested interval unscented Kalman filter (UKF) [44] is used to eliminate the possible noise in the hand gesture datasets. Moreover, due to some hand gesture datasets may contain unequal sequences from different participants, the short and long sequences should be changed into a standard sequence. The length of the standard sequence is set to a fixed value n based on both the average length of the sequence of each gesture. For short sequences, the length of them is increased by linear interpolation. For long sequences, we will eliminate the first few frames and the last few frames of the sequence because there are usually many pause actions at the beginning and the end, and they are not important to the whole gesture. The joint $P_{i,k}(t)$, a full hand skeleton $H_k(t)$ and the k^{th} class gesture G_k are shown as follows:

$$P_{i,k}(t) = \left[x_{i,k}(t), y_{i,k}(t), z_{i,k}(t)\right] \tag{8}$$

$$H_k(t) = \sum_{i=1}^{m} P_{i,k}(t) \tag{9}$$

$$G_k = \sum_{t=0}^{n} H_k(t) \tag{10}$$

where n is the scale of the k^{th} class gesture sequences; all of the joints i in one hand are combined into a full hand skeleton $H_k(t)$ when the time scale of the k^{th} class gesture is at t; m represents the maximum number of joints in a full hand skeleton; the shape of the k^{th} class gesture G_k is processed into $n \times (am)$; the feature scale is am, and a is the spatial scale.

The shape of the standard sequence is split into $n_1 \times n_2 \times (am)$ through the segmentation gestures (SG), where the k^{th} class gesture forms n_1 sets of sequences and the time scale of each set is n_2.

Then, the skeleton-based hand gesture sequences can be mapped to the same specific interval by normalizing the changing hand joints, which is effective to improve the convergence rate of our network:

$$\mu_B \leftarrow \frac{1}{m} H_k(t) \tag{11}$$

$$\sigma_B^2 \leftarrow \frac{1}{m} \sum_{i=1}^{m} \left(P_{i,k}(t) - \mu_B\right)^2 \tag{12}$$

$$\hat{P}_{i,k}(t) \leftarrow \frac{P_{i,k}(t) - \mu_B}{\sqrt{\sigma_B^2 + \varepsilon}} \tag{13}$$

where μ_B is the mean of the sample and σ_B^2 is the sample variance; The linear transformation is added to these sequences and normalizes them to obtain $\hat{P}_{i,k}(t)$, which limits the distribution of them and

makes the network more stable during training; ε is the role of the minimum number, which avoids zero in the denominator in the expression.

The network may lose its original feature representation capabilities by the normalization. A pair of learnable parameters γ and β are set for each normalization to eliminate hidden dangers, which is used to restore the original distribution to obtain $Q_{i,k}(t)$.

$$Q_{i,k}(t) \leftarrow \gamma \cdot \hat{P}_{i,k}(t) + \beta \equiv BN_{\gamma,\beta}\left(P_{i,k}(t)\right) \tag{14}$$

In the formula, $BN_{\gamma,\beta}\left(P_{i,k}(t)\right)$ is represented as a complete batch normalization (BN).

Additionally, the joint coordinates of the hand skeleton-based sequences are limited by the neighborhood, which increases the variance of the estimate and is not conducive to enhancing network learning. The average pooling layer (avgpool) can solve the above problems, which makes the structure of the skeleton-based sequence simpler and more stable, improves the calculation efficiency of the network, and avoids over-fitting during training. Here $Q_{i,k}(t)$ is introduced to represent the changes in the same joints of the adjacent multiple frames after the avgpool:

$$\hat{Q}_{i,k}(t) = \frac{1}{f^2} \sum_{t=t_0}^{t_0+2} \left(Q_{i,k}(t)\right) \tag{15}$$

$$\hat{G}_k = \sum_{t=0}^{\hat{n}} \sum_{i=1}^{m} \hat{Q}_{i,k}(t) \tag{16}$$

where f is the size of a filter of the average pooling layer; the size of $\hat{n}$ is set to the equal of $n_1 * (n_2/f)$; the shape of $\hat{G}_k$ is split into $n_1 \times (n_2/f) \times (am/f)$, which contains the features information of the k^{th} class gesture, and as the input sequence of our network.

Finally, for one-shot learning, only a small part of the hand gesture datasets was taken as the training samples for subsequent experiments.

4.2.2. Implementation

The whole process of dynamic hand gesture recognition based on one-shot learning is shown in Figure 5.

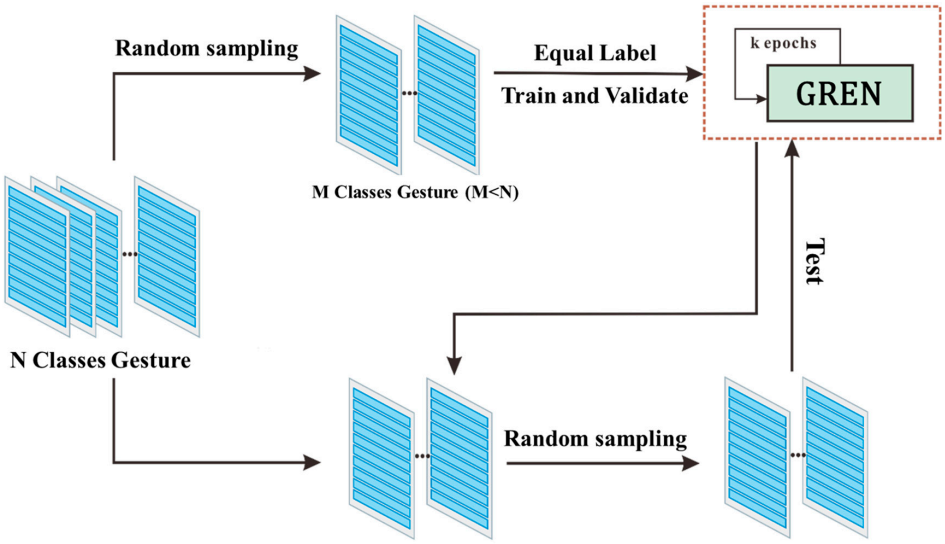

Figure 5. Flowchart of the implementation.

Firstly, the M different classes are randomly selected from the N classes already contained in the dataset, which prevents the network from simply mapping class labels to the output. From the episode to the next episode, those classes presented in the current episode with the associated labels and specific samples will be shuffled. Later, the sample sequences are equally singled out from each of the M classes, which are supposed to be of the same size. Each group from the randomly re-labeled M classes extracts 10 sets of sequences as the training data at random. Of course, it is not enough to take merely 10 sets of the sequence for each training. Additionally, the corresponding batch size is taken by random sampling as the input of training. Then, the model will be validated by the validation set every k epochs, and output the prediction accuracy with corresponding loss. Finally, the above processes are repeated until the model converges.

For the converged network model, the test set will be randomly selected to evaluate its generalization ability. After the test, the model's ability to recognize those new unrecognized sequences will be the criterion of model selection.

According to the public DHGD hand gesture dataset, the time scale is set to 60 so that the size of the gesture sequences will be at least 100 sets in each class. After the data preprocessing, the shape of $\hat{G}_k$ is split into $60 \times 20 \times 22$. For "one-shot learning", 60%, 20%, and 20% of the data are used for the training set, the validation set, and the test set, respectively.

The DHGD dataset contains two different ways of 14-classes gestures: one finger and the whole hand. N is set to 14 as the number of the unique class; M is set to 3 as the number of sample classes; k is set to 100 as the epoch-size in each training. For the 28-classes gestures encompassing the above two ways, N is set to 28 as the number of the unique class, while sizes of M and k remain unchanged in each training. A grid search [45] is performed over a number of hyper-parameters: controller size (200 hidden units for an LSTM), the learning rate ($4e-5$), the number of read–write heads from memory (4), and training times (80,000). For the 14-classes, the batch size is taken as 8, while it is set to 16 in the case of 28-classes. The model presents the best results over those hyper-parameters configurations.

In this study, another comparison experiment has been conducted based on the MSRA dataset. The time scale is also set to 12. After the data preprocessing, the shape of $\hat{G}_k$ is segmented into $60 \times 5 \times 21$. Moreover, 50% of the data is used for the training set; 25% of the data utilized for the validation set; 25% of the data applied to the test set. For the MSRA dataset containing hand gestures of 17 classes, N is set to 17 as the number of the unique classes, and sizes of M and k remain unchanged in each training. Compared with the 14-classes and 28-classes, hyper-parameters for the 17-classes are shown: controller size (200 hidden units for an LSTM), the learning rate ($4e-5$), the number of read–write heads from memory (4), batch size (16), and training times (70,000).

5. Results and Discussion

To visualize the process of the recognition accuracy measured on the validation set, we have separately analyzed two different ways of 14-classes: one finger and the whole hand, and the 28-classes encompassing both the above two ways. In addition, the accuracy curve is shown in Figure 6.

From Figure 6, the 14-classes, (1) represents right-hand gestures performed with one finger, and (2) represents gestures with the whole hand. The curve of the one-finger classified by our method is shown in blue, the curve of the whole-hand is shown with an orange line, and the curve of the 28-classes is shown with a grey line. It is observed that the recognition accuracy of the 14-classes (2) is superior to the 14-classes (1), and the 28-classes is between those two. Compared with the 14-classes (1), the 28-classes has better performance.

To assess the effectiveness of our algorithm for classifying the hand gestures of DHGD into 14-classes and 28-classes, we compare the standard LSTM network with regard to their DHGD recognition accuracy. Table 2 shows the comparison results of skeleton-based hand gesture recognition between LSTM and GREN networks.

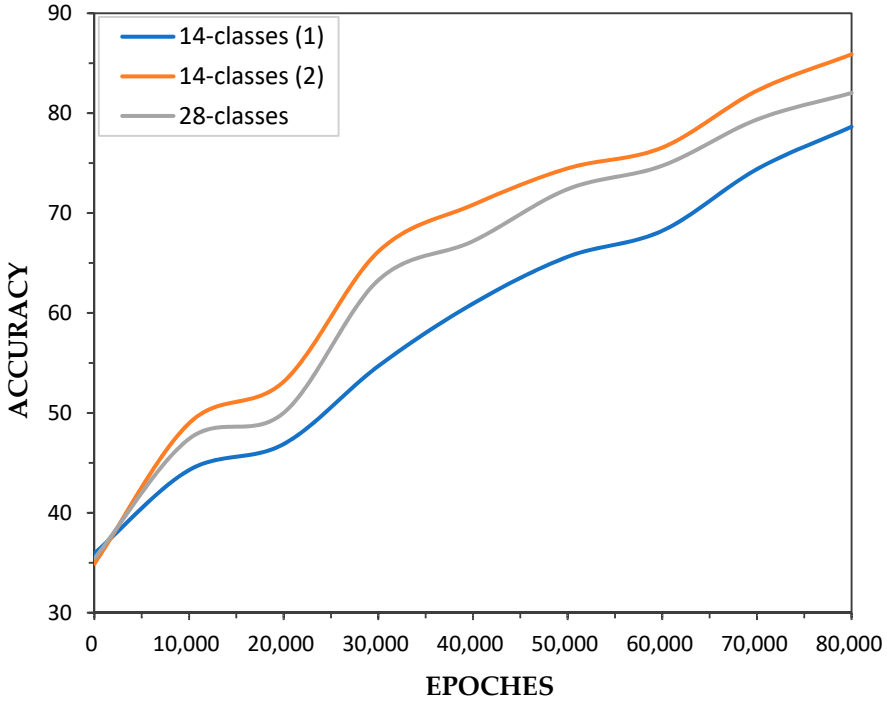

Figure 6. The accuracy curve of our method for 14-classes and 28-classes in the DHGD dataset.

Table 2. Comparison results between long short-term memory (LSTM) and gesture recognition using an enhanced network (GREN) networks based on the DHGD dataset.

Type		LSTM (%)	GREN (%)
14-classes	1	75.18	78.65
	2	79.82	85.90
28-classes	both	76.89	82.03

From Table 2, the final accuracy of our GREN network reaches 82.29% for the 14-classes classification that is the average of the two ways and 82.03% for the 28-classes classification. The proposed network indicates that recognition accuracy can reach 78.65% for the one-finger and 85.90% for the whole-hand. Thus, compared with the standard LSTM networks, the accuracy of the recognition increased by approximately 5.14%, the accuracy of the one-finger increased by approximately 3.47%, and the whole-hand accuracy increased by 6.08%, which show excellent performance of our method in one-shot learning.

We compare the GREN network with the state-of-the-art algorithm in DHGD, and the results are shown in Table 3.

For the different ways of learning, a mature scheme of one-shot learning combined with hand gesture recognition has not been proposed before. Those advanced methods of comparison adopt the way of recognizing large size samples for experiments. While our GREN network uses small size samples in the DHGD dataset and trains based on one-shot learning.

Compared with other advanced algorithms, our method also performs well. For the 14-classes classification, the final accuracy of our GREN network is 82.29%, which is higher than most other algorithms. Additionally, our GREN network presents a higher accuracy in the 28-classes recognition than does that of the other advanced algorithm. A comparison of other advanced algorithms shows that the accuracy of the GREN network will not reduce significantly with the increase of the classes of hand gestures in the 28-classes recognition. Experimental results suggest that the proposed GREN network is an efficient method for hand gesture recognition.

Table 3. Result of different method comparison for 14/28-classes gestures on dynamic hand gesture dataset using skeleton-based data.

Learning	Methods	Accuracy 14-Classes Gestures	Accuracy 28-Classes Gestures
Large-samples	HON4D: Histogram of Oriented 4D Normals for Activity Recognition from Depth Sequences [46]	75.53%	74.03%
	3-D Human Action Recognition by Shape Analysis Of Motion Trajectories on Riemannian Manifold [47]	79.61%	62.00%
	Joint Angles Similarities and HOG2 for Action Recognition [48]	80.85%	76.53%
	Key Frames with Convolutional Neural Network [18]	82.90%	71.90%
	Skeleton-Based Dynamic Hand Gesture Recognition [49]	83.07%	79.14%
	NIUKF-LSTM [44]	84.92%	80.44%
	SL-Fusion-Average [36]	85.46%	74.19%
	MFA-Net [29]	85.75%	81.04%
One-shot	GREN	**82.29%**	**82.03%**

Besides, to verify the robustness of the network, a similar experimental setup has also been performed on the MSRA hand gesture dataset. To more clearly demonstrate our network, we compared the experimental result with the LSTM network based on the MSRA dataset, which is shown in Table 4.

Table 4. Comparison results between LSTM and GREN networks based on the MSRA dataset.

Type	LSTM (%)	GREN (%)
17-classes	72.92	79.17

From Table 4, the final accuracy of our network is 79.17% for the 17-classes classification. Additionally, compared with the LSTM networks, the accuracy of the recognition increased by approximately 6.25%, which shows the better performance of the GREN network. The experiment verifies that this network could be replicated for other similar datasets, even if they are small sample size datasets.

6. Conclusions

This paper proposes the GREN network to recognize dynamic hand gestures based on a small number of skeleton-based sequence samples. According to the MANN network, the ability to store and update sequence data is further enhanced by introducing the average pooling layer (avgpool) and batch normalization (BN), so that we can combine the hand skeleton sequence with the GREN network to achieve dynamic hand gesture recognition based on one-shot learning. Experiments with the DHGD hand gesture dataset demonstrate the state-of-the-art performance of the GREN network for skeleton-based dynamic hand gesture recognition based on one-shot learning. Additionally, the MSRA hand gesture dataset verifies the robustness of our GREN network.

Author Contributions: Conceptualization, Y.Q.; methodology, A.W.; software, S.Z.; supervision, G.C.; validation, Y.Q.; writing—original draft, S.Z.; writing—review and editing, C.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Fundamental Research Funds for the Central Universities grant number 201762005, the National Natural Science Foundation of China grant number 41906155, and the Marine S&T Fund of Shandong Province for Pilot National Laboratory for Marine Science and Technology (Qingdao) grant number 2019GHZ023.

Acknowledgments: The authors gratefully acknowledge the support of the Fundamental Research Funds for the Central Universities (Grant No.: 201762005), National Natural Science Foundation of China (Grant No.: 41906155), and Marine S&T Fund of Shandong Province for Pilot National Laboratory for Marine Science and Technology, Qingdao (Grant No.: 2019GHZ023).

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, and in the decision to publish the results.

References

1. 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.
2. Lv, Z.; Halawani, A.; Feng, S.; Ur Réhman, S.; Li, H. Touch-less interactive augmented reality game on vision-based wearable device. *Pers. Ubiquitous Comput.* **2015**, *19*, 551–567. [CrossRef]
3. Liu, J.; Wang, G.; Duan, L.; Abdiyeva, K.; Kot, A.C. Skeleton-based human action recognition with global context-aware attention lstm networks. *IEEE Trans. Image Process.* **2018**, *27*, 1586–1599. [CrossRef] [PubMed]
4. Nie, Q.; Wang, J.; Wang, X.; Liu, Y. View-invariant human action recognition based on a 3d bio-constrained skeleton model. *IEEE Trans. Image Process.* **2019**, *28*, 3959–3972. [CrossRef] [PubMed]
5. Lv, Z.; Halawani, A.; Feng, S.; Li, H.; Réhman, S.U. Multimodal hand and foot gesture interaction for handheld devices. *ACM Trans. Multimed. Comput. Commun. Appl.* **2014**, *11*, 10. [CrossRef]
6. Liu, X.; Su, Y. Tracking skeletal fusion feature for one shot learning gesture recognition. In Proceedings of the International Conference on Image, Vision and Computing, Chengdu, China, 2–4 June 2017; pp. 194–200. [CrossRef]
7. Zhu, W.; Lan, C.; Xing, J.; Zeng, W.; Li, Y.; Shen, L.; Xie, X. Co-occurrence feature learning for skeleton based action recognition using regularized deep LSTM networks. In Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, Phoenix, AZ, USA, 12–17 February 2016; pp. 3697–3703.
8. 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]
9. Santoro, A.; Bartunov, S.; Botvinick, M.; Wierstra, D.; Lillicrap, T. Meta-learning with memory-augmented neural networks. In Proceeding of the International Conference on Machine Learning, New York, NY, USA, 19–24 June 2016; pp. 1842–1850.
10. Deng, L.; Yu, D. Deep learning: Methods and applications. *Found. Trends Signal Process.* **2014**, *7*, 197–387. [CrossRef]
11. Besak, D.; Bodeker, D. Hard thermal loops for soft or collinear external momenta. *J. High Energy Phys.* **2010**, *5*, 7. [CrossRef]
12. Duchi, J.; Hazan, E.; Singer, Y. Adaptive subgradient methods for online learning and stochastic optimization. *J. Mach. Learn. Res.* **2011**, *12*, 2121–2159.
13. Howard, J.; Ruder, S. Universal Language Model Fine-tuning for Text Classification. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Long Papers), Melbourne, Australia, 15–20 July 2018; pp. 328–339. [CrossRef]
14. Bengio, Y. Deep learning of representations for unsupervised and transfer learning. In Proceedings of the ICML Workshop on Unsupervised and Transfer Learning, Edinburgh, UK, 26 June–1 July 2012; pp. 17–36.
15. Kirkpatrick, J.; Pascanu, R.; Rabinowitz, N.C. Overcoming catastrophic forgetting in neural networks. *Proc. Natl. Acad. Sci. USA* **2017**, *114*, 3521–3526. [CrossRef]
16. Greve, R.; Jacobsen, E.J.; Risi, S. Evolving neural turing machines for reward-based learning. In Proceedings of the Genetic and Evolutionary Computation Conference, Denver, CO, USA, 20–24 July 2016; pp. 117–124. [CrossRef]
17. Li, Z.; Hoiem, D. Learning without forgetting. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *40*, 2935–2947. [CrossRef]
18. De Smedt, Q.; Wannous, H.; Vandeborre, J.P.; Guerry, J.; LeSaux, B.; Filliat, D. 3D hand gesture recognition using a depth and skeletal dataset: SHREC'17 track. In Proceedings of the Workshop on 3D Object Retrieval. Eurographics Association, Lyon, France, 23–24 April 2017; pp. 33–38. [CrossRef]

19. 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]
20. Tan, D.J.; Cashman, T.; Taylor, J.; Fitzgibbon, A.; Tarlow, D.; Khamis, S.; Shotton, J.; Izadi, S. Fits like a glove: Rapid and reliable hand shape personalization. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 5610–5619. [CrossRef]
21. Supančič, J.S.; Rogez, G.; Yang, Y.; Shotton, J.; Ramanan, D. Depth-based hand pose estimation: Methods, data, and challenges. *Int. J. Comput. Vis.* **2018**, *126*, 1180–1198. [CrossRef]
22. Lv, Z. Wearable smartphone: Wearable hybrid framework for hand and foot gesture interaction on smartphone. In Proceedings of the IEEE International Conference on Computer Vision Workshops, Sydney, Australia, 1–8 December 2013; pp. 436–443. [CrossRef]
23. 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]
24. Tang, D.; Taylor, J.; Kohli, P.; Keskin, C.; Kim, T.K.; Shotton, J. Opening the black box: Hierarchical sampling optimization for estimating human hand pose. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 3325–3333. [CrossRef]
25. Ye, Q.; Yuan, S.; Kim, T.K. Spatial attention deep net with partial pso for hierarchical hybrid hand pose estimation. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016; pp. 346–361. [CrossRef]
26. Guo, H.; Wang, G.; Chen, X.; Zhang, C.; Qiao, F.; Yang, H. Region ensemble network: Improving convolutional network for hand pose estimation. In Proceedings of the IEEE International Conference on Image Processing, Beijing, China, 17–20 September 2017; pp. 4512–4516. [CrossRef]
27. 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]
28. Wang, G.; Chen, X.; Guo, H.; Zhang, C. Region ensemble network: Towards good practices for deep 3d hand pose estimation. *J. Visual Commun. Image Represent.* **2018**, *55*, 404–414. [CrossRef]
29. 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]
30. 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]
31. Chin-Shyurng, F.; Lee, S.E.; Wu, M.L. Real-time musical conducting gesture recognition based on a dynamic time warping classifier using a single-depth camera. *Appl. Sci.* **2019**, *9*, 528. [CrossRef]
32. Ding, J.; Chang, C.W. An adaptive hidden Markov model-based gesture recognition approach using Kinect to simplify large-scale video data processing for humanoid robot imitation. *Multimed. Tools Appl.* **2016**, *75*, 15537–15551. [CrossRef]
33. Kumar, P.; Saini, R.; Roy, P.P.; Dogra, D.P. A position and rotation invariant framework for sign language recognition (SLR) using Kinect. *Multimed. Tools Appl.* **2018**, *77*, 8823–8846. [CrossRef]
34. Mazhar, O.; Navarro, B.; Ramdani, S.; Passama, R.; Cherubini, A. A real-time human-robot interaction framework with robust background invariant hand gesture detection. *Robot. Comput. Integr. Manuf.* **2019**, *60*, 34–48. [CrossRef]
35. 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]
36. 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]
37. Lake, B.M.; Salakhutdinov, R.; Tenenbaum, J.B. Human-level concept learning through probabilistic program induction. *Science* **2015**, *350*, 1332–1338. [CrossRef] [PubMed]
38. Snell, J.; Swersky, K.; Zemel, R. Prototypical networks for few-shot learning. *Adv. Neural Inf. Process. Syst.* **2017**, 4077–4087.
39. Koch, G.; Zemel, R.; Salakhutdinov, R. Siamese neural networks for one-shot image recognition. In Proceedings of the ICML Deep Learning Workshop, Lille, France, 10–11 July 2015; Volume 2.

40. Cai, Q.; Pan, Y.; Yao, T.; Yan, C.; Mei, T. Memory matching networks for one-shot image recognition. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018. [CrossRef]
41. Finn, C.; Abbeel, P.; Levine, S. Model-agnostic meta-learning for fast adaptation of deep networks. In Proceedings of the International Conference on Machine Learning, Sydney, Australia, 6–11 August 2017; pp. 1126–1135.
42. Ravi, S.; Larochelle, H. Optimization as a model for few-shot learning. In Proceedings of the International Conference on Learning Representations, Toulon, France, 24–26 April 2017.
43. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. In Proceedings of the International Conference on Learning Representations, San Diego, CA, USA, 7–9 May 2015.
44. 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. [CrossRef]
45. Pontes, F.J.; Amorim, G.F.; Balestrassi, P.P.; De Paiva, A.P.; Ferreira, J.R. Design of experiments and focused grid search for neural network parameter optimization. *Neurocomputing* **2016**, *186*, 22–34. [CrossRef]
46. Oreifej, O.; Liu, Z. Hon4d: Histogram of oriented 4d normals for activity recognition from depth sequences. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, 23–28 June 2013; pp. 716–723. [CrossRef]
47. Devanne, M.; Wannous, H.; Berretti, S.; Pala, P.; Daoudi, M.; Del Bimbo, A. 3-d human action recognition by shape analysis of motion trajectories on riemannian manifold. *IEEE Trans. Cybern.* **2014**, *45*, 1340–1352. [CrossRef] [PubMed]
48. 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]
49. 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. [CrossRef]