sensors

MDPI

Article

High-Density Surface EMG-Based Gesture Recognition Using a 3D Convolutional Neural Network

Jiangcheng Chen [1,*], Sheng Bi [1,2,*], George Zhang [1] and Guangzhong Cao [3]

[1] Shenzhen Academy of Robotics, Shenzhen 518057, China; gqzhang@szarobots.com
[2] School of Computer Science and Engineering, South China University of Technology, Guangzhou 510006, China
[3] Shenzhen Key Laboratory of Electromagnetic Control, Shenzhen University, Shenzhen 518060, China; gzcao@szu.edu.cn
* Correspondence: jiangcheng.0502@163.com (J.C.); bisheng@szarobots.com (S.B.)

Received: 8 January 2020; Accepted: 19 February 2020; Published: 21 February 2020

Abstract: High-density surface electromyography (HD-sEMG) and deep learning technology are becoming increasingly used in gesture recognition. Based on electrode grid data, information can be extracted in the form of images that are generated with instant values of multi-channel sEMG signals. In previous studies, image-based, two-dimensional convolutional neural networks (2D CNNs) have been applied in order to recognize patterns in the electrical activity of muscles from an instantaneous image. However, 2D CNNs with 2D kernels are unable to handle a sequence of images that carry information concerning how the instantaneous image evolves with time. This paper presents a 3D CNN with 3D kernels to capture both spatial and temporal structures from sequential sEMG images and investigates its performance on HD-sEMG-based gesture recognition in comparison to the 2D CNN. Extensive experiments were carried out on two benchmark datasets (i.e., CapgMyo DB-a and CSL-HDEMG). The results show that, where the same network architecture is used, 3D CNN can achieve a better performance than 2D CNN, especially for CSL-HDEMG, which contains the dynamic part of finger movement. For CapgMyo DB-a, the accuracy of 3D CNN was 1% higher than 2D CNN when the recognition window length was equal to 40 ms, and was 1.5% higher when equal to 150 ms. For CSL-HDEMG, the accuracies of 3D CNN were 15.3% and 18.6% higher than 2D CNN when the window length was equal to 40 ms and 150 ms, respectively. Furthermore, 3D CNN achieves a competitive performance in comparison to the baseline methods.

Keywords: high-density surface EMG (HD-sEMG); finger gesture recognition; deep learning; convolutional neural network (CNN)

1. Introduction

Prosthetic hands that are capable of performing various movements have been, from a mechanical point of view, remarkably improved since the last decade. However, the lack of an effective control interface still prevents its practical application in amputees. Surface electromyography (sEMG) is the non-invasive electrical recording of muscle activity and provides access to neural information associated with human movement. In comparison to touch screens and keyboards, the sEMG-based interface could offer a natural and intuitive way of controlling for disabilities. From its inception until now, the myoelectric prosthesis has been designed with trigger control, proportional control, and pattern recognition-based control [1]. In the pattern recognition-based control approach, a classifier trained with supervised learning was employed to map sEMG activity to one of the predefined classes that correspond to different control commands. In the past decades, many methods have been proposed

to design a sEMG pattern recognition-based interface, some of which have achieved high accuracy with many classes in a laboratory environment [2–4].

According to the number and the configuration of electrodes, the acquired surface EMG can be divided into sparse multi-channel sEMG and high-density sEMG (HD-sEMG) [5–8]. With regard to the multi-channel case, great achievements have been made and recognition accuracy has been shown to reach up to 95% in some research [1]. However, the exact positioning of the electrodes (a mostly bipolar configuration) is required to collect the right signals and the signal is considered to represent the activity of the whole muscle. Indeed, the effects of the placement of electrodes were studied in [9]. In other words, the information provided by the sparse multi-channel sEMG is highly dependent on the positioning of electrodes. Another disadvantage of the sparse multi-channel sEMG is that the fault of one channel can cause a reduced performance. In contrast, the HD-sEMG collected by the 2D electrode grids can provide both spatial and temporal information, and have the potential to overcome the aforementioned shortcomings. In previous studies, the concepts of the sEMG image and sEMG map have been proposed for gesture recognition. The sEMG image was spatially composed of instant values of raw HD-sEMG data according to the arrangement of the electrodes, while the sEMG map was composed of values of time windowed features [7,8,10–13]. At present, the number of electrodes contained in an electrode array could range from 32 to 350, which results in a big dataset with a high sample frequency. While HD-sEMG-based interfaces increase the production cost and computation demand, it is commonly believed that these issues in hardware can be easily resolved with the advancement of sensor devices and special micro-processing technology.

Generally, feature extraction and classifier design are two key factors involved in the development of pattern-recognition-based control interfaces. Thus far, various features have been identified in time, frequency, and time–frequency domains [1,14,15]. Angkoon et al. [14] presented 37 time domain and frequency domain features, and investigated their classification performance. Then, they also evaluated 50 time domain and frequency domain features to improve the robustness of myoelectric pattern recognition [15]. In many studies, multi-features have been used to improve the accuracy of hand gesture/motion classification. For the design of the classifier, the commonly used, conventional classification algorithms include linear discriminant analysis (LDA) [16], hidden Markov models (HMM) [17], support vector machines (SVM) [18], adaptive neuro-fuzzy inference system (ANFIS) [19], and so on. By optimizing the selection of features and classifiers, satisfactory results can be obtained offline. Therefore, the selection of features and classifiers is highly problem-dependent and requires strong professional knowledge, which limits the application of these methods.

Compared to feature engineering and conventional shallow learning models, deep learning models, which contain many hidden layers, can automatically learn a hierarchy of features from raw data and have recently made a huge impact on pattern recognition [20,21]. The deep learning approach is also used in the field of EMG processing [22]. Mukhopadhyay and Samui conducted an empirical exploration on the deep-learning-model-based sEMG signal classification and demonstrated that their methods could outperform other classifiers such as the k-nearest neighbor, random forest, and decision tree [23]. Zhang et al. [24] applied the deep learning method long short-term memory (LSTM) to classify hand gestures by multimodal data collected from the inertial measurement unit, Myo armband (Thalmic Labs Inc.), and pressure sensors. Convolutional neural networks (CNNs) have been successfully used. In the beginning, CNNs were used together with feature pre-extracting (i.e., feature engineering). Allard et al. [25] presented a specific CNN architecture with spectrograms as the input to identify seven hand/wrist gestures. Zhai et al. [26] proposed a CNN-based classifier using a dimension-reduced spectrogram as the input and updated itself to maintain its performance over a long time. In contrast to using the pre-extracted sEMG features as the input, Geng et al. [11] found that there were patterns inside instantaneous HD-sEMG images (formed with raw data) and presented a CNN model to recognize the gestures with instantaneous HD-sEMG images. They later proposed a hybrid CNN and RNN (recurrent neural network) architecture to take advantage of the time series information [27]. Inspired by the correlation between certain muscles and each specific

gesture, they also proposed a two-stage multi-stream CNN to enhance recognition accuracy [28]. This method decomposes the original sEMG image into many equal-sized streams and independently learns representative features by 2D CNN.

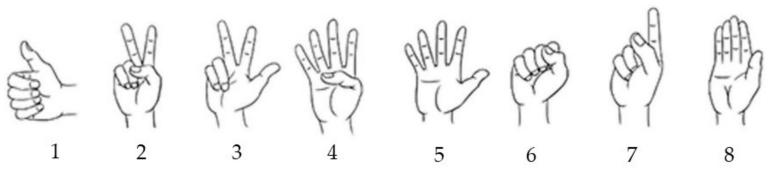
In summary, HD-sEMG signals that carry both spatial and temporal information of muscle activity and employ CNNs as classifiers are becoming increasingly studied in developing sEMG-based interfaces. However, as a class of deep models, CNNs with 2D convolution kernels are limited to handling a single image that only carries spatial information. In contrast, 3D CNN, with 3D convolution kernels, is capable of extracting features from both spatial and temporal dimensions. The first 3D CNN was proposed by Ji et al. [29] to analyze video data, and achieved a superior performance in comparison to the baseline methods. Besides this, Al-Hammadi et al. [30] employed transfer learning of a 3D CNN for hand gesture recognition by using video data and achieved excellent accuracy. In fact, HD-sEMG signals are sequential data like video data. Motivated by this, we investigated a CNN with 3D convolution kernels for HD-sEMG pattern recognition in this paper. To the best of our knowledge, this is the first comparative study of 2D CNN and 3D CNN in HD-sEMG-based gesture recognition.

The rest of the paper is organized as follows: The materials and methods including the data used in this paper, the construction of the 3D CNN, and the experiments of HD-sEMG-based finger gesture recognition are described in Section 2. The experimental results are presented and discussed in Section 3. The paper is concluded in Section 4.

2. Materials and Methods

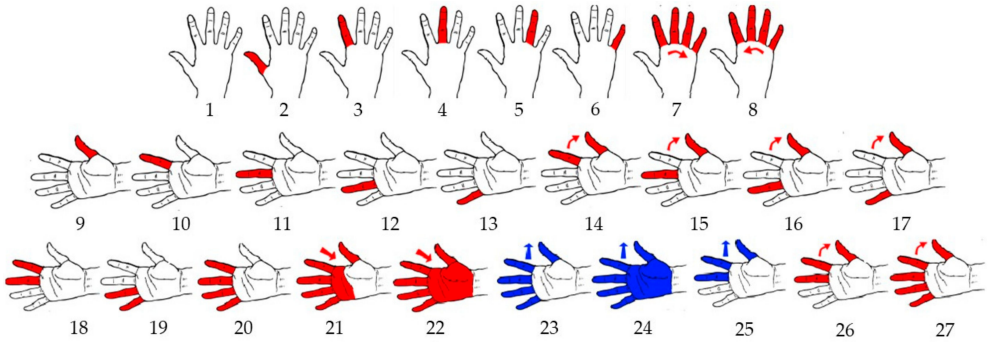
2.1. Data and Pre-Processing

CapgMyo and CSL-HDEMG are two HD-sEMG benchmark databases and were utilized in this study. Both of them are available online [11,12]. For CapgMyo, we downloaded the sub-database named DB-a, which contains eight isometric and isotonic finger gestures obtained from 18 healthy able-bodied subjects. The eight finger gestures are shown in Figure 1a. Each subject performed 10 trials for each gesture. Each gesture was held for 3 to 10 seconds. The HD-sEMG signals in CapgMyo DB-a were recorded with an 8×16 electrode grid wrapped around the right forearm. The 128-channel signals were band-pass filtered at 20–380 Hz and sampled at 1000 Hz. The CSL-HDEMG database contained 27 finger gestures obtained from five subjects. The 27 finger gestures covered the extension and flexion of individual fingers and incorporated some typical gestures that might be used in human–computer interaction. These were organized in three sets in [12], which were tapping gestures, bending gestures, and multi-finger gestures, respectively. The signals were bipolarly recorded by a 7×24 electrode grid that led to a total of 168 channels of usable data by removing the channels of data from the difference of the last electrode in one column and the first electrode in the next column. The HD-sEMG signals in CSL-HDEMG were sampled at 2048 Hz and each trial was recorded for three seconds. Each subject was asked to perform one gesture within a time interval of three seconds. Each subject recorded five sessions and performed 10 trials for each gesture in every session. The gestures in CSL-HDEMG and their indexes are shown in Figure 1b.



(**a**)

Figure 1. *Cont.*



(**b**)

Figure 1. Iconic illustrations of eight gestures contained in CapgMyo DB-a and 27 gestures contained in SCL-HDEMG. (**a**) Eight finger gestures. These are (1) thumb up, (2) extension of index and middle finger while flexing the others, (3) flexion of the ring and little fingers while extending the others, (4) thumb opposing base of the little finger, (5) abduction of the fingers, (6) fingers flexed together in a fist, (7) pointing index, and (8) adduction of extended fingers, respectively. (**b**) Twenty-seven finger gestures. (1) idle gesture, (2–6) tap once with the finger shown in red, (7–8) four fingers are tapped sequentially in the direction of the arrow, (9–13) bending of the finger, (14–17) pinch grips with the thumb, (18–20) bending of two or four fingers, (21) making a fist without applying force, (22) making a fist with force, (23–25) stretching out the fingers marked in blue, and (26–27) pressing the marked fingers against the thumb.

For the purposes of this study, first, the sequential sEMG images should be obtained. For CapgMyo DB-a, there were two datasets: the preprocessed dataset and the raw dataset. We selected the preprocessed dataset in which the power-line interference has been removed and the values of the signal have been normalized to [−1, 1]. Furthermore, the sEMG signals of each trial in the preprocessed dataset contain 1000 frames of data that correspond to the static part of finger movement. In other words, for each trial, the middle one-second data have been segmented. Therefore, in our work, we only transformed the values of the signals to [0, 1] linearly and reshaped the data of each frame to form an 8 × 16 sEMG image. As a result, 1000 sequential sEMG images were generated for each finger gesture recording per subject. Figure 2 shows the partial, sequential sEMG images obtained from one selected trial. From the figure, we can see that the instantaneous image evolved with time, although they corresponded to a static part of the finger movement.

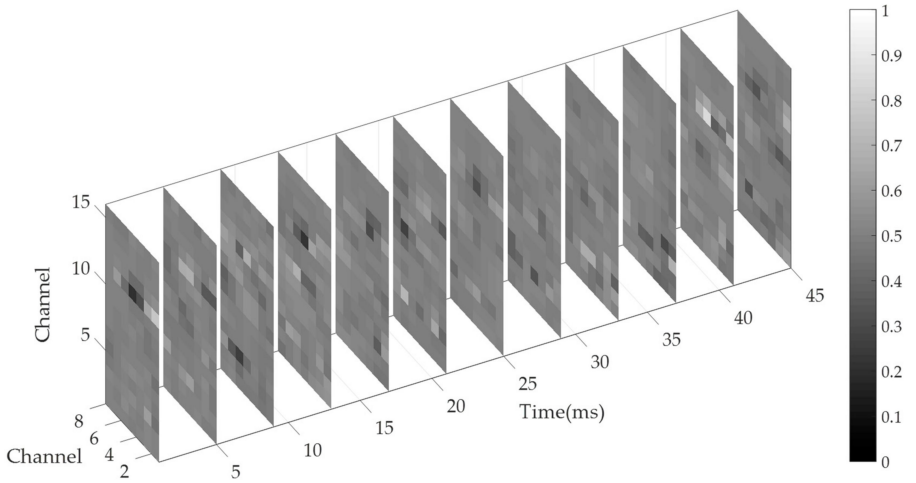


Figure 2. The partial of sequential surface electromyography (sEMG) images obtained from one selected trial of CapgMyo DB-a. Signals were sampled in time at 1000 frames/s. Twelve instantaneous images (out of 45, interval = 4 ms) are shown in the figure. Each pixel of the image represents one channel. The brighter the pixel, the stronger the electrical activity of the channel.

For CSL-HDEMG, more processing was required to obtain the sequential sEMG images. Like the experiments of Amma et al. [12], the sEMG signals were first filtered by a zero-lag fourth-order Butterworth filter with a pass-band of 20–400 Hz to remove signal artifacts. Then, full-wave rectification, followed by a low-pass zero-lag Butterworth filter with a 4 Hz cutoff frequency were applied on the denoised signals and the time series of the intensities were acquired [31]. Next, the thresholding approach, on a sliding window without overlap, was applied on the time series of the intensity to search for the segment containing activity. The length of the sliding window was set to 150 sampling points (73.2 ms) [12], resulting in the length of the segmented muscle activity being a multiple of 150. Instead of calculating the root-mean-square (RMS) in the work by Amma et al. [12], the average intensity value on every window was calculated for each channel and the average of the summarized values of all channels per window was chosen as the threshold for the segmentation in our work. Based on the above process, the activity segments for all 168-channel sEMG signals in every trial were identified. Finally, the values of the denoised signals were transformed to linear and sequential sEMG images [0,1], corresponding to the 27 gestures that were generated for each trial. The segmented sequential sEMG images included continuous finger movement instead of just static movement, like that seen in CapgMyo DB-a.

2.2. Construction of 3D Convolutional Neural Network

2.2.1. 3D Convolution

Encouraged by the achievements in the field of image recognition, CNNs have been extended to be three-dimensional using 3D kernels to make them suitable for video sequences or volumetric medical imaging data [29,32]. In 2D CNNs, units in a convolution layer are organized in planes within which all the units share the same set of weights to perform 2D convolution on the local neighborhood of a feature map in the previous layer. The area of the local neighborhood is called the receptive field of the unit and its size is the size of a 2D convolution kernel. A complete convolution layer is composed of several planes and the outputs of each plane generate a feature map. Compared to 2D CNN, 3D convolution is achieved in 3D CNN by convolving a 3D kernel to the cube formed by stacking multiple contiguous frames together. The 3D convolution on a cube will produce another cube. As a result, 3D CNNs perform convolution spatiotemporally and capture both the spatial and temporal structures, while 2D CNNs perform convolution spatially and only capture spatial information. Formally, the value at position (x, y, z) in the jth cube in the ith layer is given by

$$v_{ij}^{xyz} = f\left(b_{ij} + \sum_{m} \sum_{p=0}^{P_i-1} \sum_{q=0}^{Q_i-1} \sum_{r=0}^{R_i-1} \omega_{ijm}^{pqr} v_{(i-1)m}^{(x+p)(y+q)(z+r)}\right) \quad (1)$$

where P_i, Q_i, and R_i are the height, width, and length of the kernel, respectively. m is the index of the set of cubes in the previous layer. ω_{ijk}^{pqr} is the (p, q, r)th of the kernel connected to the kth cube in the previous layer. v_{ij}^{xyz} designates the output of the jth cube in the ith layer at the (x, y, z) position. b_{ij} is the bias. $f(\cdot)$ represents the activation function, which in most cases is a sigmoid function, tan hyperbolic, or rectified linear unit (RELU).

2.2.2. 3D CNN Architecture

In this section, we present an ordinary 3D CNN architecture that was used in this paper. While several studies on 3D CNNs have been conducted, there has been no rigorous theoretical analysis in designing the architecture of the network. Usually the depth and the kernel size are task-dependent. The first 3D CNN model, proposed by Ji et al. [29], consisted of one hardwired layer, three convolution layers, two sub-sampling layers, and one fully connected layer. Tran et al. [33] proposed a 3D CNN architecture with eight convolutions, five max-pooling, and two fully connected layers that were designed to learn spatiotemporal features from a video dataset. Their findings suggest that a

homogeneous architecture with small $3 \times 3 \times 3$ convolution kernels could achieve the best performance. A 3×3 kernel is also employed for the 2D CNN model, which is often used for gesture recognition based on instantaneous surface EMG images [11].

The finally determined 3D CNN architecture is illustrated in Figure 3, based on previous studies, and was the purpose of our exploration. The network contained one input layer, three 3D convolutional layers, two pooling layers, two fully connected layers, and ended with a G-way fully connected layer and a *softmax* function, where G is the number of gestures to be classified. The input layer is the first layer and specifies the size of the input. We refer to the input cube with a size of $l \times h \times w \times c$, where l is the number of frames, c is the number of channels ($c = 1$ due to the grayscale of sEMG image and the c will be omitted in the following text), and h and w are the height and width of each image, respectively. The 3D kernel size is indicated by $d \times k \times k$, where d is the temporal depth and k is the spatial size. The number of kernels in the three convolutional layers were 32, 64, and 64, respectively. The size of all kernels was set to $3 \times 3 \times 3$ and the convolution stride was fixed to $1 \times 1 \times 1$. Zero padding with a size of $0 \times 1 \times 1$ was applied for the second and third convolutional layers, followed by a pooling layer with a size of $s \times 1 \times 1$, where the choice of s was related to the sampling rate of the sEMG signals in this study. This meant that sub-sampling was only performed temporally. The two fully connected layers consisted of 512 and 128 neurons, respectively. The rectified linear unit (RELU) activation function was applied for all the hidden layers.

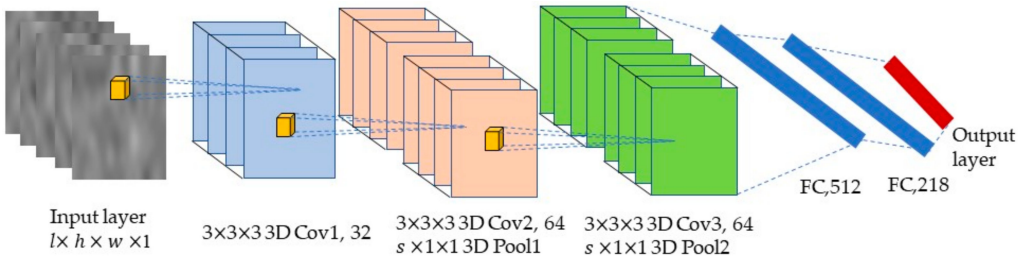


Figure 3. The architecture of the proposed 3D convolutional neural network. The network consists of an input layer, three convolutional layers, two pooling layers, two fully connected layers, and an output layer. The input of the network is a clip of a sequence of sEMG images and is expressed as $l \times h \times w \times 1$, where l is the number of continuous sEMG images, h and w are the height and width of each image, respectively, and 1 indicates that the sEMG images are grayscale images. The outputs of the network are the classes of the finger gestures.

2.2.3. Network Training

The proposed 3D CNN was implemented based on the platform of Keras 2.3.0, which is a high-level neural network application programming interface (API), written in Python and capable of running on top of TensorFlow. The workstation we employed included a Nvidia GTX 1080 Ti GPU and one Intel Core i7 CPU. The stochastic gradient descent algorithm was used to update the parameters. The initial learning rate was set to 0.1 and was divided by two if there was no loss in improvement after 10 iterations. During the training stage, the following cross-entropy loss function was utilized to optimize the output

$$\text{Loss} = -\frac{1}{n}\sum_{i=1}^{n} y_i \log \overline{y}_i + (1 - y_i) \log\left(1 - \overline{y}_i\right) \quad (2)$$

where y_i and $\overline{y}_i$ are the sample labels and the actual output, respectively, and n is the number of input samples. In addition, batch normalization before each non-linearity layer and dropout with a probability of 0.5 on the two fully connected layers were employed to optimize the network, where the former can accelerate the network training and the latter can prevent the network from over-fitting [34,35].

2.3. Majority Voting

With the trained 3D network, the patterns for each input cube could be recognized. As described above, the length of the cube could range from one to many frames. In order to satisfy the real-time usage constraints, the recognition time should be shorter than 300 ms [27]. In fact, multiple input cubes could be generated in a 300 ms time period. Therefore, simple majority voting over multiple consecutive recognition results could be used to enhance the recognition accuracy. The voting window needed to be less than 300 ms. Considering that, in a voting window, the recognition results of a neural network for consecutive cubes are M_i $(i = 1, 2 \ldots, n_c)$ and the target gesture is T, then the recognition accuracy P can be calculated as

$$P = \frac{m}{n_c} \times 100\% \tag{3}$$

where m is the number of recognition results that satisfy $M_i = T$. What should be emphasized here is that the unit of the voting window in this work is in ms. In other words, if the length of the input cube, which was equal to the length of the sliding window for segmentation, was t_{cube} ms and there were n_c consecutive cubes in a voting window, then the length of the voting window was defined as $n_c \cdot t_{cube}$ ms.

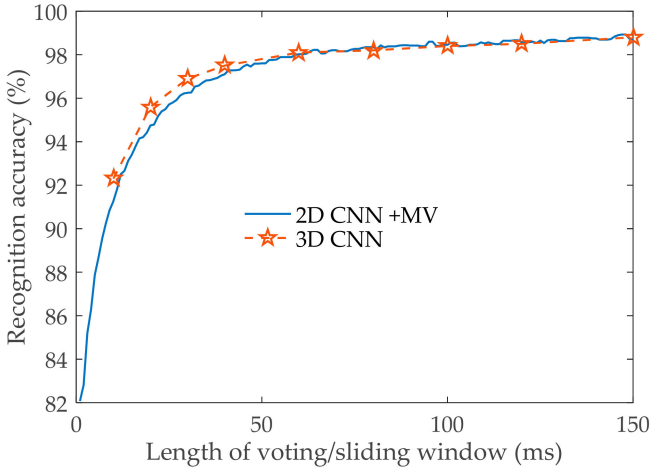
2.4. Experiments

The proposed model was evaluated on the two benchmark databases described in Section 2.1. The CapgMyo DB-a database was divided into two parts of the same size for training and testing, as described in [11]. The sequential sEMG images of each trial were split into small segments using the sliding window strategy and a number marked each sEMG cube as one of the gestures. The cube size was $l \times 8 \times 16$. Considering that a large number of labeled samples were required for network learning, the sliding window using the overlap method for EEG data augmentation was applied on CapgMyo DB-a [36]. What needs to be emphasized here is that only the data from the training part adopted the data augmentation strategy (with a stride two frames) during segmentation. Additionally, the size of the pooling layer was set to $2 \times 1 \times 1$. For the CSL-HDEMG database, the leave-one-out cross-validation scheme was performed in each session on every subject at the same time in [12], in which each of the 10 trials served as testing data while the remaining nine trials constituted the training data. Similarly, the sequential sEMG images of each trial were split into small clips by sliding windows. The cube size was $l \times 7 \times 24$ and the size of the pooling layer was set to $4 \times 1 \times 1$. In the experiments, we investigated the recognition accuracies with different lengths l of the sliding window. For comparison purposes, we constructed a 2D CNN to perform recognition after supervised learning. As far as we know, except for the form of convolutions, the predictive capability of the model was also related to the structure and depth of the network. Therefore, in order to make a fair and effective comparison, the three 3D convolutional layers in the proposed network were replaced by three 2D convolutional layers. The temporal dimensions of the kernels for convolution, padding, and pooling were removed, while the size of the spatial dimensions remained the same. Together with the majority voting strategy over consecutive frames, the recognition accuracies of 2D CNNs, with different voting window lengths, were evaluated on the two databases. Furthermore, the statistics of recognition accuracies were compared with the results from previous studies.

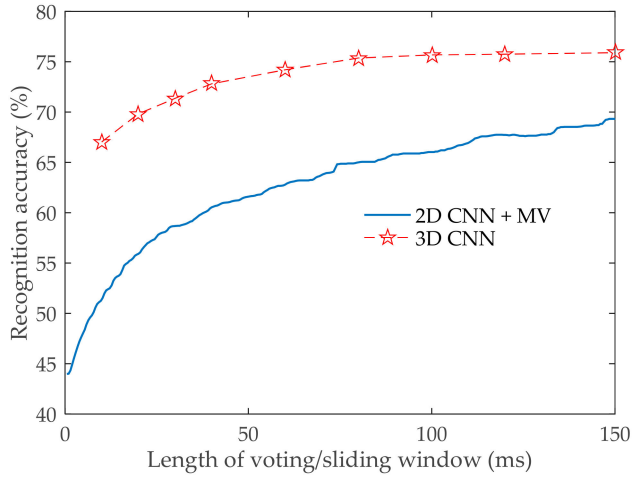
3. Results and Discussion

Figure 4 shows the comparison of recognition accuracies between the 3D and 2D CNNs proposed in this paper; Figure 4a shows the statistical results from the CapgMyo DB-a database, while Figure 4b shows the statistical results from the CSL-HDEMG database. In Figure 4a, the blue solid line represents the recognition accuracies over the first six subjects by using 2D CNN, together with the majority voting (2D CNN + MV) method with respect to different lengths of voting windows. It can be seen that the recognition accuracy increases with increasing lengths of voting windows, where the accuracies corresponding to the lengths of 10 ms and 150 ms were 91.3% and 98.7%, respectively.

The pentagram connection indicates the recognition accuracies over the first six subjects by using 3D CNN with different lengths l of the sliding window for segmentation, where the lengths at the location of the pentagrams were 10, 20, 30, 40, 60, 80, 100, 120, and 150 frames, respectively. The accuracies corresponding to the lengths of 10 and 150 were 92.3% and 98.9%, respectively. It was found that the recognition accuracy of the 3D CNN was higher than the 2D CNN + MV when the length of the sliding/voting window was less than 60, and was basically the same when the length of the window was higher. In Figure 4b, the blue solid line represents the recognition accuracies of subject1–session1 by using 2D CNN + MV, while the pentagram connection represents the recognition accuracies by using 3D CNN. Figure 4b shows that the accuracy obtained by both methods increased with an increase in the length of the window, which was the same as the case using CapgMyo DB-a. It was also found that the recognition accuracies by using 3D CNN were significantly higher than by using 2D CNN + MV, in comparison to the case using CapgMyo DB-a. The accuracies corresponding to the lengths of 10 ms (20 frames) and 150 ms (307 frames) were 51.4% and 70.4% with 2D CNN + MV and were 67.0% and 75.9% with 3D CNN.



(**a**)



(**b**)

Figure 4. The comparation of recognition accuracies between 3D CNN over different sliding windows and 2D CNNs over different voting windows. (**a**) Recognition accuracies over the first 6 subjects of the CapgMyo DB-a database. (**b**) Recognition accuracies of subject1–session1 of the CSL-HDEMG database.

In contrast, the improvement in the accuracy of CSL-HDEMG was significantly greater than that of CapgMyo DB-a. This is partly because the improvement space is limited due to CapgMyo DB-a being close to saturation, but the most important reason is that only the static part of the finger movement is

included for CapgMyo DB-a, while the dynamic part is also included in CSL-HDEMG. Therefore, the 3D CNN method, capable of extracting dynamic change information, has a great advantage. It can also be seen from Figure 4 that the increase in accuracy was not obvious after a certain length for the 3D CNN. Therefore, it is not worth selecting a long length for the input cube of the 3D CNN because with the increase in length, the number of weights needing to be learnt increases rapidly, leading to a higher computation cost. The situation in this paper is shown in Table 1. In contrast, the number of weights needing to be learnt for the 2D CNN, in the cases of CapgMyo DB-a and CSL-HDEMG, were 2.90×10^6 and 3.77×10^6, respectively.

Table 1. The number of weights needing to be learnt over different lengths l of the sliding window for segmentation (i.e., the length of the input cube).

l(ms)	10	20	30	40	60	80	100	120	150
Number$(\times 10^6)$ [1]	3.08	8.71	17.11	22.75	36.79	50.80	64.86	78.87	101.29
Number$(\times 10^6)$ [2]	4.05	7.88	15.36	19.20	26.88	38.18	49.50	57.18	72.31

[1] In the case of the CapgMyo DB-a database. [2] In the case of the CSL-HDEMG database.

Based on the above findings, we can conclude that, in the case of the same network architecture, 3D convolution can achieve better performance than the combination of the 2D convolution and majority voting, especially for the sEMG data, which contain distinctly dynamic processes. However, it is necessary to make a trade-off between accuracy and computation load. The computation load is directly related to the length of the window for segmentation. Therefore, we will investigate the combined use of the 3D CNN with a short length and the majority voting strategy (3D CNN + MV) below. From the perspective of computation load, the lengths of the sliding windows chosen in this work for segmentation were 10 and 20 frames for CapgMyo DB-a and CSL-HDEMG, respectively. Table 2 shows the recognition accuracies over the entire database by using 3D CNN + MV. The results of 2D CNN + MV are also presented in the table. First, it shows that the recognition accuracy of the 3D CNN + MV method was higher than the 2D CNN + MV method. For CapgMyo DB-a, the accuracy of 3D CNN + MV corresponded to the voting length of 40 ms and 150 ms and were 95.5% and 98.6%, respectively, which was lower than the results (99.0% and 99.6%, respectively) presented in [11]. However, for CSL-HDEMG, the accuracy of 3D CNN + MV corresponded to a voting length of 150 ms and was 90.7%, which was higher than the 89.3% presented in [11]. This further confirms the advantages of 3D CNN in the pattern recognition of EMG signals with dynamic processes.

Table 2. The recognition accuracies over the entire database with different lengths l_V of voting windows.

l_V (ms)	40	50	60	70	80	90	100	110	120	130	140	150
2D CNN + MV (%) [1]	94.5	95.3	96.0	96.2	96.4	96.4	99.6	96.7	96.7	96.8	96.8	97.1
3D CNN + MV (%) [1]	95.5	96.7	97.2	97.8	97.9	98.0	98.3	98.3	98.4	98.4	98.6	98.6
2D CNN + MV (%) [2]	61.7	62.7	64.1	64.9	65.8	67.1	67.7	67.8	69.6	69.6	71.0	72.1
3D CNN + MV (%) [2]	77.0	79.1	81.1	83.4	84.6	84.7	86.5	86.9	87.9	89.5	90.6	90.7

[1] In the case of CapgMyo DB-a database. [2] In the case of CSL-HDEMG database.

The recognition accuracies in Table 2 are a global measure. In order to compare the classification rate of individual gestures between different approaches, and considering the imbalance in the dataset results from the activity interval segmentations, the four accumulated confusion matrixes, corresponding to the voting length of 150 ms (in Table 2) over all recognition results, were calculated as examples of all the approaches taken, as shown in Figure 5. The values in the matrixes were rounded. From Figure 5a, we can find that the lowest classification accuracy is gesture 6, which is often confused with gesture 5. However, in the case of using the 3D CNN + MV method, the accuracy rate of gesture 6 increased to 98.4% and this gesture showed no confusion with gesture 5. Furthermore, the accuracy

rates for gestures 4–8 all increased, while gestures 1–3 basically remained the same. From Figure 5c,d, we can see that there were many gestures with accuracy rates between 50% and 60% using the 2D CNN + MV method, while the lowest individual accuracy rate was 73.6% (gesture 1, which is often confused with gestures 2, 3, and 9) by using the 3D CNN + MV method. We can also see from the number distributions of the two figures that confusion occurred more frequently with 2D CNN + MV than with 3D CNN + MV. The above findings confirm the better performance of the 3D CNN.

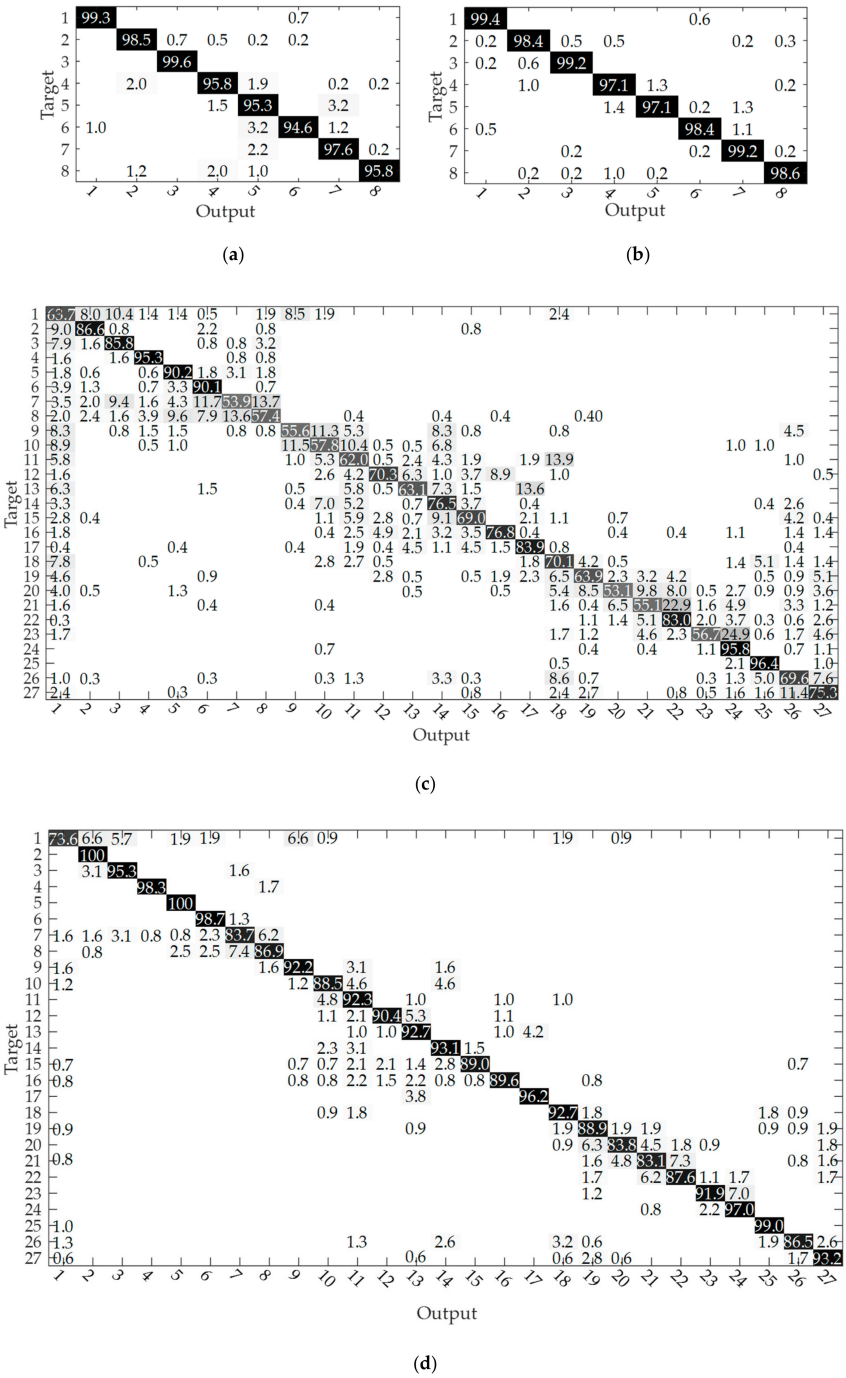


Figure 5. Accumulated confusion matrixes of each method for both databases corresponding to the voting length 150 ms. The higher the percentage value, the darker the color. (**a**) The case of 2D CNN + MV for CapgMyo DB-a. (**b**) The case of 3D CNN + MV for CapgMyo DB-a. (**c**) The case of 2D CNN + MV for CSL-HDEMG. (**d**) The case of 3D CNN + MV for CSL-HDEMG.

We also counted the results of simple majority voting over the entire segment of each trial for CSL-HDEMG by using the 3D CNN + MV method. The accuracy reached 95.3%, which was higher than the 90.4% presented in [12], and close to the 96.7% presented in [11] and 96.1% presented in [25]. Figure 6 shows the accumulated confusion matrix over all subjects and sessions by using 3D CNN + MV with an input of size 20 × 7 × 24 and majority voting over the entire segment of each trial. From the figure, we can see that gesture 1 (i.e., idle gesture) is the most confused gesture due to its confusion with gestures 2, 3, and 9. This is consistent with the results in [12]. Gesture 1 had the lowest rate at only 80.3%, while some gestures had 100% accuracies. Gestures 7 and 8 were often confused, in addition to gestures 23 and 24. We could also see that gesture 21 was frequently confused with gestures 20 and 22. This may be due to the similarity between these confusing gestures such as gestures 7 and 8, gestures 23 and 24 and gestures 21 and 22.

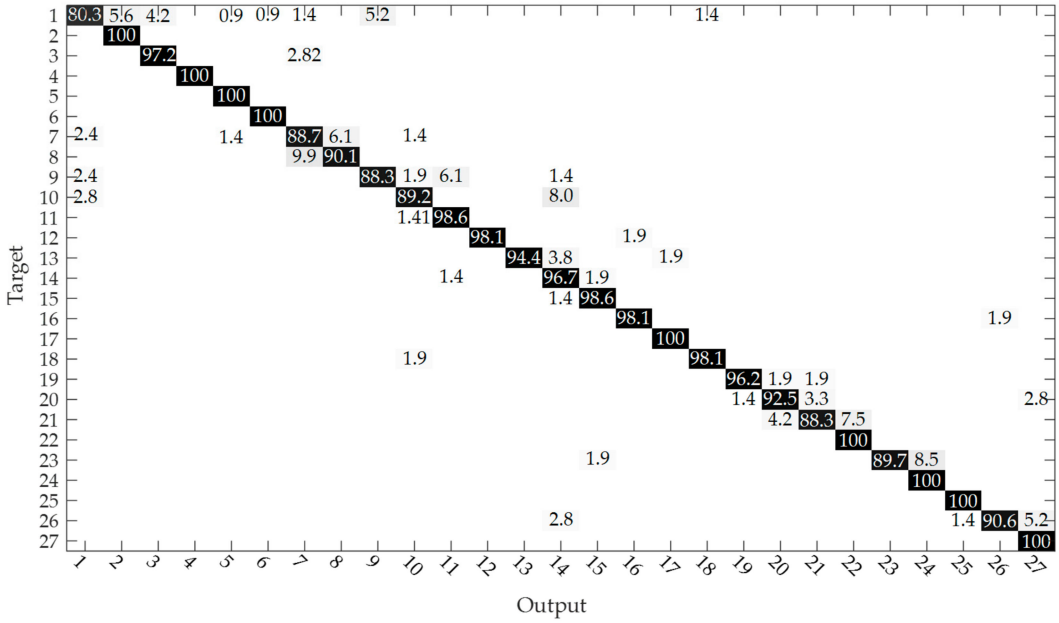


Figure 6. Accumulated confusion matrix over all subjects and sessions. The higher the percentage value, the darker the color.

4. Conclusions

In this study, we constructed a 3D convolutional neural network (3D CNN) to perform high-density surface EMG (HD-sEMG) based gesture recognition. Based on the two benchmark datasets (i.e., CapgMyo DB-a and CSL-HDEMG), the performance of 3D CNNs as well as its comparation to 2D CNN were thoroughly investigated. From the above experimental studies and analysis, we can obtain the following conclusions. First, in the case of the same network architecture, 3D convolution can achieve a better performance than the combination of 2D convolution and majority voting, especially for the sEMG data that contain the dynamic part of the finger movement. This benefit comes from the capability of 3D CNN to capture both the temporal and spatial information of muscle activity from high-density surface EMG signals collected by a grid of electrodes. Second, when the length of the input clip increases, the time taken to learn and recognize the 3D CNN increases quite rapidly. Therefore, a trade-off between recognition accuracy and computation load should be made. Additionally, by using the 3D CNN, together with the majority voting strategy, a competitive performance, in comparison to the baseline methods, can be achieved. In summary, HD-sEMG based gesture recognition, by using 3D CNN, could be a promising solution to develop muscle interfacing for prosthetic control. In addition, the 3D CNN architecture used in this work is relatively simple in comparison to the deep neural network used in other studies. In future research, we will try to explore newer and deeper

neural network architectures and add 3D convolution operations to further improve the accuracy of HD-sEMG based gesture recognition.

Author Contributions: Conceptualization, J.C. and S.B.; Methodology, J.C.; Software, J.C. and S.B.; Validation, G.Z. and G.C.; Formal analysis, G.Z. and G.C.; Investigation, J.C.; Resources, J.C. and S.B.; Data curation, J.C. and G.Z.; Writing—original draft preparation, J.C.; Writing—review and editing, G.Z.; Visualization, J.C.; Supervision, G.Z.; Project administration, S.B.; Funding acquisition, S.B. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China under Grant No. U1813214 and the Shenzhen Basic Research Projects Foundation under Grant No. JCYJ 20160429161539298.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Hakonen, M.; Piitulainen, H.; Visala, A. Current state of digital signal processing in myoelectric interfaces and related applications. *Biomed. Signal Process. Control* **2015**, *18*, 334–359. [CrossRef]
2. Xing, K.; Yang, P.; Huang, J.; Wang, Y.; Zhu, Q. A real-time EMG pattern recognition method for virtual myoelectric hand control. *Neurocomputing* **2014**, *136*, 345–355. [CrossRef]
3. Mccool, P.; Petropoulakis, L.; Soraghan, J.J.; Chatlani, N. Improved pattern recognition classification accuracy for surface myoelectric signals using spectral enhancement. *Biomed. Signal Process. Control* **2015**, *18*, 61–68. [CrossRef]
4. Duan, F.; Dai, L.; Chang, W.; Chen, Z.; Zhu, C.; Li, W. sEMG-based identification of hand motion commands using wavelet neural network combined with discrete wavelet transform. *IEEE Trans. Ind. Electron.* **2016**, *63*, 1923–1934. [CrossRef]
5. Atzori, M.; Gijsberts, A.; Castellini, C.; Caputo, B.; Mittaz Hager, A.-G.; Elsig, S.; Giatsidis, G.; Bassetto, F.; Müller, H. Electromyography data for non-invasive naturally-controlled robotic hand prostheses. *Sci. Data* **2014**, *1*, 140053. [CrossRef] [PubMed]
6. Shim, H.M.; Lee, S. Multi-channel electromyography pattern classification using deep belief networks for enhanced user experience. *J. Cent. South Univ.* **2015**, *22*, 1801–1808. [CrossRef]
7. Monica, R.M.; Miguel, A.M.; Joan, F.A. High-density surface EMG maps from upper-arm and forearm muscles. *J. Neuroeng. Rehabil.* **2012**, *9*, 85.
8. Atzori, M.; Cognolato, M.; Müller, H. Deep learning with convolutional neural networks applied to electromyography data: A resource for the classification of movements for prosthetic hands. *Front. Neurorobot.* **2016**, *10*, 9. [CrossRef]
9. Huebner, A.; Faenger, B.; Schenk, P.; Scholle, H.C.; Anders, C. Alteration of Surface EMG amplitude levels of five major trunk muscles by defined electrode location displacement. *J. Electromyogr. Kinesiol.* **2015**, *25*, 214–223. [CrossRef]
10. Rojas-Martnez, M.; Mañanas, M.A.; Alonso, J.F.; Merletti, R. Identification of isometric contractions based on high density EMG maps. *J. Electromyogr. Kinesiol.* **2013**, *23*, 33–42. [CrossRef]
11. Geng, W.; Du, Y.; Jin, W.; Wei, W.; Hu, Y.; Li, J. Gesture recognition by instantaneous surface EMG images. *Sci. Rep.* **2016**, *6*, 36571. [CrossRef] [PubMed]
12. Amma, C.; Krings, T.; Böer, J.; Schultz, T. Advancing muscle-computer interfaces with high-density electromyography. In Proceedings of the 33rd annual ACM Conference on Human Factors in Computing Systems, Seoul, Korea, 18–23 April 2015; pp. 929–938.
13. Ghaderi, P.; Marateb, H.R. Muscle activity map reconstruction from high density surface EMG signals with missing channels using image inpainting and surface reconstruction methods. *IEEE Trans. Biomed. Eng.* **2017**, *64*, 1513–1523. [CrossRef] [PubMed]
14. Angkoon, P.; Pornchai, P.; Chusak, L. Feature reduction and selection for EMG signal classification. *Expert Syst. Appl.* **2012**, *39*, 7420–7431.
15. Angkoon, P.; Franck, Q.; Sylvie, C.; Christine, S.; Franck, T.-B.; Yann, L. EMG feature evaluation for improving myoelectric pattern recognition robustness. *Expert Syst. Appl.* **2013**, *40*, 4832–4840.
16. Tkach, D.; Huang, H.; Kuiken, T.A. Study of stability of time-domain features for electromyographic pattern recognition. *J. Neuroeng. Rehabil.* **2010**, *7*, 21. [CrossRef]

17. Chan, A.D.C.; Englehart, K.B. Continuous myoelectric control for powered prostheses using hidden markov models. *IEEE Trans. Biomed. Eng.* **2004**, *52*, 121–124. [CrossRef]
18. Oskoei, M.A.; Hu, H. Support vector machine-based classification scheme for myoelectric control applied to upper limb. *IEEE Trans. Biomed. Eng.* **2008**, *55*, 1956–1965. [CrossRef]
19. Khezri, M.; Jahed, M. A neuro-fuzzy inference system for sEMG-based identification of hand motion commands. *IEEE Trans. Ind. Electron.* **2011**, *58*, 1952–1960. [CrossRef]
20. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444. [CrossRef]
21. Shrestha, A.; Mahmood, A. Review of deep learning algorithms and architectures. *IEEE Access* **2019**, *7*, 53040–53065. [CrossRef]
22. Angkoon, P.; Erik, S. EMG pattern recognition in the era of big data and deep learning. *Big Data Cogn. Comput.* **2018**, *2*, 21.
23. Mukhopadhyay, A.K.; Samui, S. An experimental study on upper limb position invariant EMG signal classification based on deep neural network. *Biomed. Signal Process. Control* **2020**, *55*, 101669. [CrossRef]
24. Zhang, X.; Yang, Z.; Chen, T.; Chen, D.; Huang, M.-C. Cooperative sensing and wearable computing for sequential hand gesture recognition. *IEEE Sens. J.* **2019**, *19*, 5775–5783. [CrossRef]
25. Allard, U.C.; Nougarou, F.; Fall, C.L.; Giguère, P.; Gosselin, C.; Laviolette, F.; Gosselin, B. A convolutional neural network for robotic arm guidance using sEMG based frequency-features. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Daejeon, Korea, 9–14 October 2016; pp. 2464–2470.
26. Zhai, X.; Jelfs, B.; Chan, R.H.M.; Tin, C. Self-recalibrating surface EMG pattern recognition for neuroprosthesis control based on convolutional neural network. *Front. Neurosci.* **2017**, *11*, 279. [CrossRef] [PubMed]
27. Hu, Y.; Wong, Y.; Wei, W.; Du, Y.; Kankanhalli, M.; Geng, W. A novel attention-based hybrid CNN-RNN architecture for sEMG-based gesture recognition. *PLoS ONE* **2018**, *13*, e0206049. [CrossRef] [PubMed]
28. Wei, W.; Wong, Y.; Du, Y.; Hu, Y.; Kankanhalli, M.; Geng, W. A multi-stream convolutional neural network for sEMG-based gesture recognition in muscle-computer interface. *Pattern Recognit. Lett.* **2019**, *119*, 131–138. [CrossRef]
29. Ji, S.; Xu, W.; Yang, M.; Yu, K. 3D convolutional neural networks for human action recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2013**, *35*, 221–231. [CrossRef]
30. Al-Hammadi, M.; Muhammad, G.; Abdul, W.; Alsulaiman, M.; Hossain, M.S. Hand gesture recognition using 3D-CNN model. *IEEE Consum. Electron. Mag.* **2020**, *9*, 95–101. [CrossRef]
31. Chen, J.; Zhang, X.; Cheng, Y.; Xi, N. Surface EMG based continuous estimation of human lower limb joint angles by using deep belief networks. *Biomed. Signal Process. Control* **2018**, *40*, 335–342. [CrossRef]
32. Shan, H.; Zhang, Y.; Yang, Q.; Kruger, U.; Kalra, M.K.; Sun, L.; Cong, W.; Wang, G. 3-D convolutional encoder-decoder network for low-dose CT via transfer learning from a 2-D trained network. *IEEE Trans. Med. Imaging* **2018**, *37*, 1522–1534. [CrossRef]
33. Tran, D.; Bourdev, L.; Fergus, R.; Torresani, L.; Paluri, M. Learning spatiotemporal features with 3D convolutional networks. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015; pp. 4489–4497.
34. Ioffe, S.; Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Proceedings of the 32nd International Conference on Machine Learning, Lille, France, 6–11 July 2015; pp. 448–456.
35. Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* **2014**, *15*, 1929–1958.
36. Jiao, Z.; Gao, X.; Wang, Y.; Li, J.; Xu, H. Deep convolutional neural networks for mental load classification based on EEG data. *Pattern Recognit.* **2018**, *76*, 582–595. [CrossRef]