

## Article

# A Hand-Modeled Feature Extraction-Based Learning Network to Detect Grasps Using sEMG Signal

Mehmet Baygin <sup>1</sup>, Prabal Datta Barua <sup>2,3,4</sup>, Sengul Dogan <sup>5</sup>, Turker Tuncer <sup>5</sup>, Sefa Key <sup>6</sup>,  
U. Rajendra Acharya <sup>7,8,9,\*</sup> and Kang Hao Cheong <sup>10,\*</sup>

- <sup>1</sup> Department of Computer Engineering, College of Engineering, Ardahan University, Ardahan 75000, Turkey; mehmetbaygin@ardahan.edu.tr
  - <sup>2</sup> School of Business (Information System), University of Southern Queensland, Toowoomba, QLD 4350, Australia; prabal.barua@usq.edu.au
  - <sup>3</sup> Faculty of Engineering and Information Technology, University of Technology Sydney, Sydney, NSW 2007, Australia
  - <sup>4</sup> Cogninet Brain Team, Cogninet Australia, Sydney, NSW 2010, Australia
  - <sup>5</sup> Department of Digital Forensics Engineering, College of Technology, Firat University, Elazig 23119, Turkey; sdogan@firat.edu.tr (S.D.); turkertuncer@firat.edu.tr (T.T.)
  - <sup>6</sup> Department of Orthopedics and Traumatology, Bingöl State Hospital, Ministry of Health, Bingöl 12000, Turkey; sefa\_key@hotmail.com
  - <sup>7</sup> Department of Electronics and Computer Engineering, Ngee Ann Polytechnic, Singapore 599489, Singapore
  - <sup>8</sup> Department of Biomedical Engineering, School of Science and Technology, SUSS University, Singapore 599494, Singapore
  - <sup>9</sup> Department of Biomedical Informatics and Medical Engineering, Asia University, Taichung 41354, Taiwan
  - <sup>10</sup> Science, Mathematics and Technology Cluster, Singapore University of Technology and Design, Singapore 487372, Singapore
- \* Correspondence: aru@np.edu.sg (U.R.A.); kanghao\_cheong@sutd.edu.sg (K.H.C.)



**Citation:** Baygin, M.; Barua, P.D.; Dogan, S.; Tuncer, T.; Key, S.; Acharya, U.R.; Cheong, K.H. A Hand-Modeled Feature Extraction-Based Learning Network to Detect Grasps Using sEMG Signal. *Sensors* **2022**, *22*, 2007. <https://doi.org/10.3390/s22052007>

Academic Editor: Ivan Andonovic

Received: 21 January 2022

Accepted: 21 February 2022

Published: 4 March 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

**Abstract:** Recently, deep models have been very popular because they achieve excellent performance with many classification problems. Deep networks have high computational complexities and require specific hardware. To overcome this problem (without decreasing classification ability), a hand-modeled feature selection method is proposed in this paper. A new shape-based local feature extractor is presented which uses the geometric shape of the frustum. By using a frustum pattern, textural features are generated. Moreover, statistical features have been extracted in this model. Textures and statistics features are fused, and a hybrid feature extraction phase is obtained; these features are low-level. To generate high level features, tunable Q factor wavelet transform (TQWT) is used. The presented hybrid feature generator creates 154 feature vectors; hence, it is named Frustum154. In the multilevel feature creation phase, this model can select the appropriate feature vectors automatically and create the final feature vector by merging the appropriate feature vectors. Iterative neighborhood component analysis (INCA) chooses the best feature vector, and shallow classifiers are then used. Frustum154 has been tested on three basic hand-movement sEMG datasets. Hand-movement sEMG datasets are commonly used in biomedical engineering, but there are some problems in this area. The presented models generally required one dataset to achieve high classification ability. In this work, three sEMG datasets have been used to test the performance of Frustum154. The presented model is self-organized and selects the most informative subbands and features automatically. It achieved 98.89%, 94.94%, and 95.30% classification accuracies using shallow classifiers, indicating that Frustum154 can improve classification accuracy.

**Keywords:** frustum pattern; Frustum154; sEMG signal classification; grasp detection

## 1. Introduction

Electromyography (EMG) is a diagnostic procedure to assess the health of muscles and the nerve cells that control them (motor neurons). There are two types of EMG:

intramuscular and surface [1–3]. Intramuscular EMG is recorded with the help of invasive electrodes. With surface EMG, on the other hand, noninvasive electrodes are used to detect the electrical signals of the muscle. Surface EMG is widely preferred to detect muscle activation time and density [4]. Electromyography (EMG) signals can be used to diagnose neuromuscular diseases through muscles and nerve cells that control muscles [5,6]. These nerve cells, known as motor neurons, transmit electrical signals that cause the muscle to contract and relax. These electrical signals may be recorded using different techniques, e.g., EMG signals obtained with the help of electrodes connected to surface, such as the hands and arms, or needles/wires connected to muscle tissue [7].

EMG signals are used in clinical applications to assist in the creation of devices such as prosthetic hands/arms [8–10]. Prosthetic hands/arms have been developed for amputees, disabled people, and patients with movement loss [11,12]. These devices can improve patient quality of life; however, they are also very costly. Current control systems and applied methodologies have improved over the years in terms of increasing the mobility of these devices. Many studies on the development of artificial intelligence-assisted myoelectric control-based smart devices have been presented [13]. Systems with a myoelectric interface can interact with devices and individuals. Thus, devices developed with systems involving myoelectric interfaces provide more efficient interactions [14]. The primary purpose of these devices is to provide patients with realistic and highly efficiency movement. In this respect, EMG signals must be interpreted and processed correctly. Many machine learning techniques have been developed for the automatic and effective processing of such signals. These methods are varied to ensure that the devices operate with high efficiency, and that the signal is interpreted correctly [15–18].

Many studies are present in the literature on the analysis of EMG signals to reduce expert dependence and minimize human error [19–21]. Menon et al. [22] developed a classification technique for forearm prosthetic devices. The technique uses EMG data describing seven hand gestures. The data were collected from 9 healthy individuals and 13 amputees. The study emphasized that the classification technique was of great importance for myoelectric prosthesis. The authors constructed five cases using EMG signals with lengths of 50 ms, 150 ms, 250 ms, 350 ms, and 450 ms. Moreover, they used a linear discriminant analysis (LDA) classifier, and their maximum classification accuracy was 95.44% using EMG signals with a length of 450 ms. Mukhopadhyaya and Samui [11] proposed a method to efficiently control prosthetic devices. The approach used the Deep Neural Network method to process the EMG signal. Khushaba [23] utilized a dataset containing eight hand gestures to test the performance of the method. In their model, they used an EMG dataset collected from five participants. A deep classifier (DNN (deep neural network)) yielded 98.88% classification accuracy. Waris et al. [24] evaluated the classification performance of EMG signals obtained from healthy individuals using upper limb prostheses. Seven-day data of the individuals was evaluated in the study. The EMG signal dataset was collected from eight transradial amputees and ten healthy participants. The authors used classifiers such as artificial neural network (ANN), tree, and LDA classifiers. Their presented machine learning model achieved over 90% accuracy using ANN. Chada et al. [25] proposed a method to provide robotic control using surface EMG signals. In their method, subbands of the signal were obtained by using Tunable-Q factor wavelet transform. The dataset was collected from five subjects. Various properties were obtained from each subband; these features were then classified using a radial basis function support vector machine (SVM), with an accuracy of 97.74%. Wang et al. [26] proposed an approach that could be used in rehabilitation devices for individuals with disabilities in their upper-limbs. They used a deep model to achieve high classification rates, reaching 92% accuracy using their recurrent deep model for six class classification. Their dataset contained EMG signals from 10 healthy subjects. Arteaga et al. [27] proposed an EMG signal-based approach for the modeling and analysis of hand movements of healthy individuals. The proposed approach used machine learning methods to evaluate EMG signals. In this study, six hand movements were selected, and data from 20 individuals were

used. The classification results showed that the most successful results in the analysis of the data were obtained by k nearest neighbor (kNN). The accuracy rate was calculated as 98% for kNN. Pancholi and Joshi [28] presented a system for evaluating EMG signals which aims to recognize the structure of arm movements from EMG signals received from amputees. In the study, six different movements from four individuals were collected and evaluated. The highest accuracy rate was 97.75%, which was achieved using LDA classifier with hold-out validation. Jia et al. [29] proposed a method for classifying EMG signals using convolutional neural networks. The windowing method was used to improve the performance of the proposed method. In the utilized dataset, sEMG signals from eight participants were used; the accuracy rate was 99.38%. Tuncer et al. [4] proposed an approach to ensure proper hand movement with EMG signals, and data from nine transradial amputee patients were used. Discrete wavelet transform and ternary pattern were selected as feature extractors in their study. The evaluation results were presented according to the following parameters: accuracy (99.14%), geometric mean (99.13%), precision (99.14%), and F1-score (99.14%), employing kNN classifier with 10-fold cross-validation. Simãoa et al. [30] developed a model for the classification of EMG signals obtained from forearm muscles. In the model, feature extraction was provided by recurrent neural networks. In addition, the study was compared with long short-term memory networks and gated recurrent unit methods. Time and accuracy rates were presented using DualMyo [31] and NinaPro DB5 [32] datasets. Their model achieved about 95% accuracy using DualMyo datasets [31], and 91% using NinaPro DB5 [32] EMG datasets.

sEMG signal classification is an important research topic for machine learning and biomedical engineering. In this work, we propose a hand-modeled learning method for sEMG signal classification with high performance. To achieve our goal, three sEMG datasets were used for testing. A new effective learning method was also applied to achieve high classification ability with linear time complexity with sEMG signals; this learning model was named Frustum154. The main aim of the Frustum154 model is to select the most valuable subbands to generate features. Frustum154 comprises three main phases: (i) feature extraction using the presented frustum pattern, statistical features, and multiple parameter-based, tunable Q actor wavelet transform (TQWT) [33] decomposition, (ii) iterative neighborhood component analysis (INCA) [34] selector, and (iii) classification using a support vector machine (SVM) [35,36] or kNN [37]. Frustum154 allows us to propose a systematic hand-crafted method. It can also choose the most appropriate model for signal classification problems.

The key novelties and contributions of this model are given below:

Novelties:

- Shapes can be used to propose new local textural feature generators. Therefore, the frustum shape is used to present a new textural feature creation function, named the “frustum pattern”. By using the frustum pattern, a shape-related, graph-based local feature extraction methodology is investigated in this work.
- A new learning network called Frustum154 is presented in this paper. Frustum154 is a self-organized learning feature extraction method which uses two types of feature selection. In the feature generation/creation phase, the best features are chosen using a loss function. By using this function, Frustum154 automatically selects the best subbands for the problem.

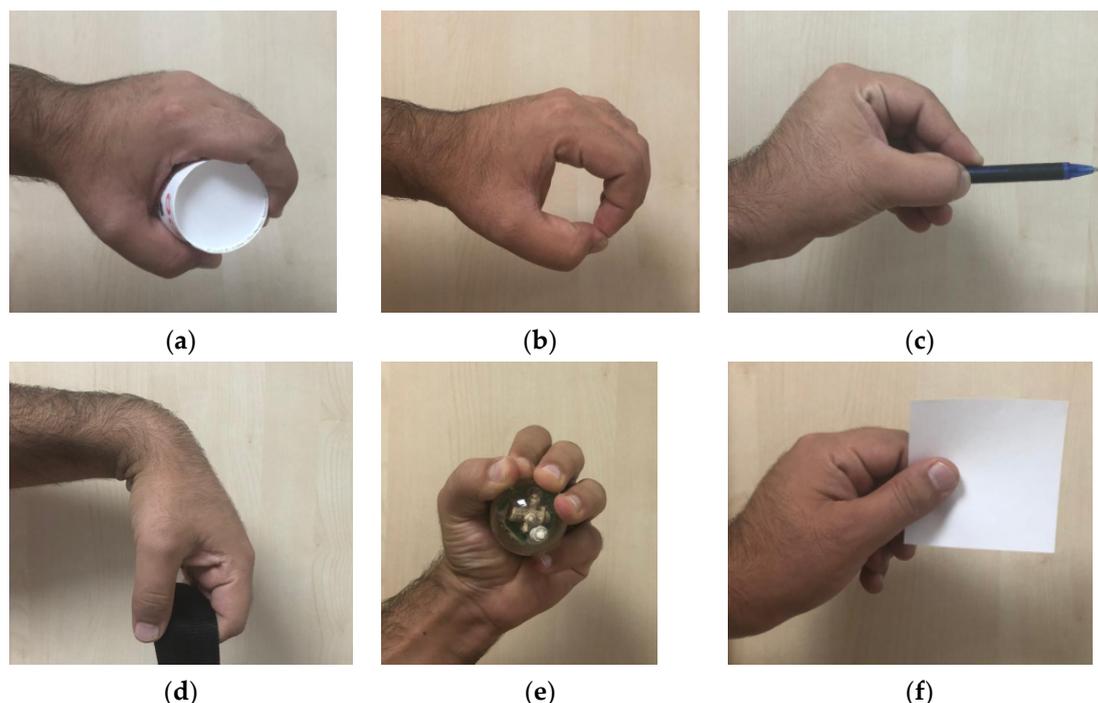
Main contributions:

- sEMG signal classification is an important signal processing topic for machine learning; deep learning models have been widely used to classify sEMG signals, achieving excellent accuracy. However, deep models are highly complex. Frustum154 is a hand-crafted, feature-based learning method which can choose the most appropriate model for signal classification problems.
- In order to demonstrate the universal classification ability of the suggested model, three sEMG signal datasets were used; the proposed model achieved over 94% classification for these datasets.

## 2. Material

### 2.1. Material

In this study, an sEMG for basic hand movement dataset from the UC Irvine Machine Learning Repository was used. This dataset contains two subdatasets, from which a third dataset was obtained by fusing the two. More details of these databases are given below. The main purpose of this dataset is to detect six basic hand-movements; (a) Cylindrical (C), (b) Tip (T), (c) Palmar (P), (d) Hook (H), (e) Spherical (S), (f) Lateral (L). Images of these gestures are presented in Figure 1 [38,39].



**Figure 1.** Definition of the proposed Prismatoid pattern. (a) Cylindrical grasp—Keeping cylindrical objects. (b) Tip—Keeping small objects. (c) Palmar—Grasping with palm. (d) Hook—Supporting a heavy load. (e) Spherical—Keeping spherical objects. (f) Lateral—Keeping thin, flat objects.

#### 2.1.1. First sEMG Dataset

The first dataset, named DB1, consisted of data from three healthy female and two healthy male subjects. The ages of subjects range from 20–22 years. These subjects performed six movements. Each subject was asked to perform each of movement for 6 seconds, and movements were repeated 30 times. Thus, 180 pieces of six-seconds, two-channel EMG signals were recorded. This dataset included a total of 900 sEMG signals. The sampling rate of the EMG signal was 500 Hz.

#### 2.1.2. Second sEMG Dataset

The second dataset (DB2) included three days of data from a healthy, male, 22-year-old subject. This subject performed 100 movements for three days. The length of the used sEMG segments was five seconds. This dataset included a total of 1800 sEMG signals. The sampling frequency was 500 Hz, as in the first dataset.

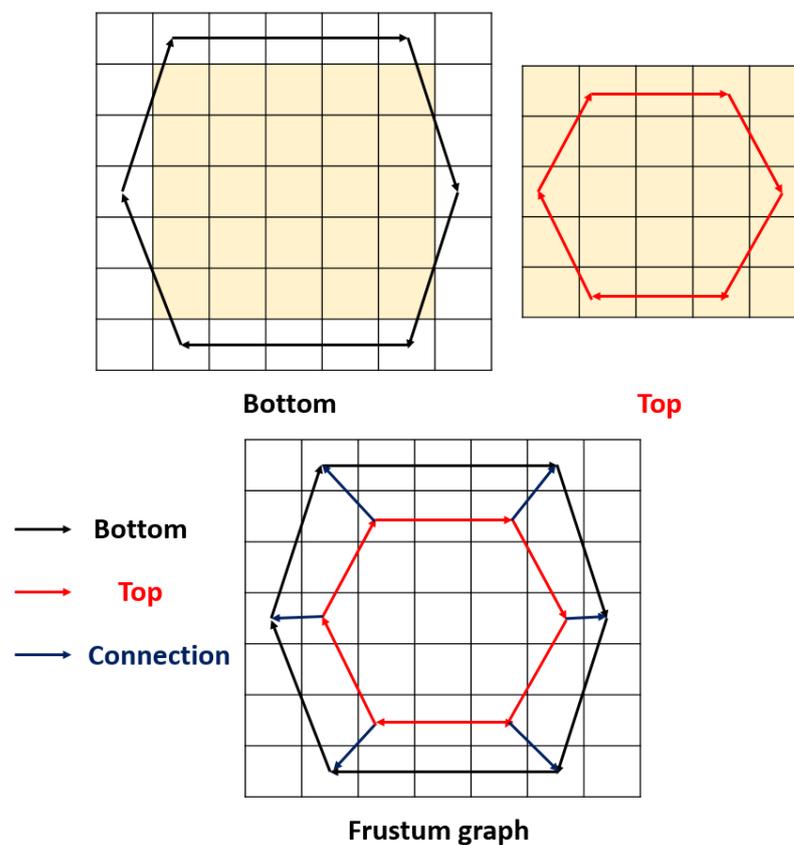
#### 2.1.3. Third sEMG Dataset

DB3 is the fused dataset. It was created by merging DB1 and DB2. It is a homogeneous dataset, with each class containing 450 sEMG signals. Therefore, DB3 contains 2700 sEMG signals in total. In this version, we created a new, large dataset by using both the first and second datasets together.

### 3. Frustum Pattern

Graph-based learning models are very popular in machine learning applications as they can solve difficult problems with a high level of accuracy. Therefore, the effects of such methods should be analyzed. We proposed a hand-modeled learning method using a graph-based feature extractor. This extractor is a graph-based function; the primary objective of this research is to investigate the feature extraction ability of the frustum shape [40] in order to create a new local textural feature generator. The proposed feature extractor was applied to three sEMG datasets to test the feature extraction ability. The main aim of the proposed frustum pattern is to extract hidden patterns from sEMG signals. The vertex and edges of the frustum [40] shape were used to create a new graph, which was utilized as the pattern for the extractor.

This shape was modeled as a feature extraction function. The frustum shape consists of two hexagons. The big hexagon is the bottom of the frustum and the small one is the top. There are six connection edges between the top and the bottom of the hexagons. Therefore, we used two matrices to model this shape as a graph-based pattern. The created bottom and top matrices and the matrices-based patterns are shown in Figure 2.



**Figure 2.** The bottom, top, and frustum graphs used to create the frustum pattern.

Figure 2 shows that the proposed frustum pattern uses  $7 \times 7$ -sized matrices, as well as two types of edges to generate binary features, i.e., bottom, top, and connection edges. Moreover, the ternary function was utilized as a kernel to generate features. The equations of the ternary bit extractors are given in Equations (1)–(3).

$$t^1(a, s) = \begin{cases} 0, & a - s \leq d \\ 1, & a - s > d \end{cases} \quad (1)$$

$$t^2(a, s) = \begin{cases} 0, & a - s \geq -d \\ 1, & a - s < -d \end{cases} \quad (2)$$

$$d = \frac{std(S)}{2} \quad (3)$$

where  $t^1(.,.)$  is the upper ternary function,  $t^2(.,.)$  is the lower ternary function,  $a, s$  denote input parameters,  $d$  is the threshold,  $std(.)$  defines standard deviation function and  $S$  defines the utilized input signal.

The steps of the proposed frustum pattern are:

**1:** Divide the signal into overlapping blocks with a length of 49.

$$obl(j) = S(i + j - 1), i \in \{1, 2, \dots, Len - 48\}, j \in \{1, 2, \dots, 49\} \quad (4)$$

where  $obl$  represents the overlapping block and  $i, j$  are indices.

**2:** Create a matrix with a size of  $7 \times 7$  using  $obl$ .

$$mat(k, l) = obl(c), c \in \{1, 2, \dots, 49\}, k \in \{1, 2, \dots, 7\}, l \in \{1, 2, \dots, 7\} \quad (5)$$

where  $mat$  is  $7 \times 7$  sized matrix to apply frustum pattern.

**3:** Generate bits by applying the frustum pattern and ternary bit extractors.

$$\begin{pmatrix} bit_k^b(1) \\ bit_k^b(2) \\ bit_k^b(3) \\ bit_k^b(4) \\ bit_k^b(5) \\ bit_k^b(6) \end{pmatrix} = t^k \begin{pmatrix} mat(1, 2), mat(1, 6) \\ mat(1, 6), mat(4, 7) \\ mat(4, 7), mat(7, 6) \\ mat(7, 6), mat(7, 2) \\ mat(7, 2), mat(4, 1) \\ mat(4, 1), mat(1, 2) \end{pmatrix}, k \in \{1, 2\} \quad (6)$$

$$\begin{pmatrix} bit_k^t(1) \\ bit_k^t(2) \\ bit_k^t(3) \\ bit_k^t(4) \\ bit_k^t(5) \\ bit_k^t(6) \end{pmatrix} = t^k \begin{pmatrix} mat(2, 3), mat(2, 5) \\ mat(2, 5), mat(4, 6) \\ mat(4, 6), mat(6, 5) \\ mat(6, 5), mat(6, 3) \\ mat(6, 3), mat(4, 2) \\ mat(4, 2), mat(2, 3) \end{pmatrix} \quad (7)$$

$$\begin{pmatrix} bit_k^c(1) \\ bit_k^c(2) \\ bit_k^c(3) \\ bit_k^c(4) \\ bit_k^c(5) \\ bit_k^c(6) \end{pmatrix} = t^k \begin{pmatrix} mat(1, 2), mat(2, 3) \\ mat(1, 6), mat(2, 5) \\ mat(4, 7), mat(4, 6) \\ mat(7, 6), mat(6, 5) \\ mat(7, 2), mat(6, 3) \\ mat(4, 1), mat(4, 2) \end{pmatrix} \quad (8)$$

where  $bit_k^b$ ,  $bit_k^t$  and  $bit_k^c$  are the  $k^{\text{th}}$  bottom, top, and connection bits.

**4:** Calculate map signal by transforming bits to decimal numbers.

$$map_k^b(i) = \sum_{j=1}^6 bit_k^b(j) \times 2^{j-1} \quad (9)$$

$$map_k^t(i) = \sum_{j=1}^6 bit_k^t(j) \times 2^{j-1} \quad (10)$$

$$map_k^c(i) = \sum_{j=1}^6 bit_k^c(j) \times 2^{j-1} \quad (11)$$

**5:** Extract histograms of the six map signals. Each histogram has 64 elements.

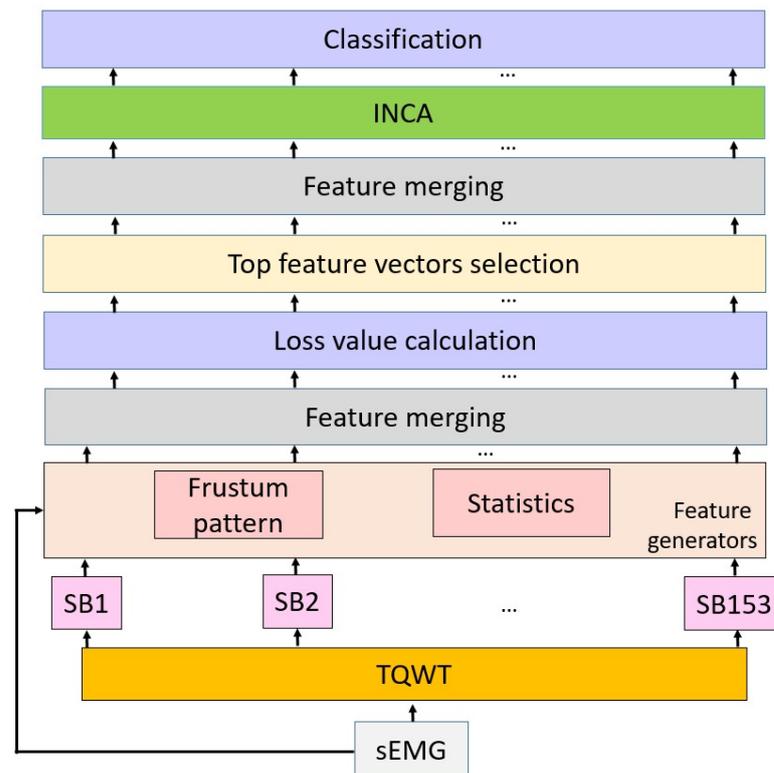
**6:** Merge the extracted histograms to create a feature vector with a length of 384.

$$fv(f + 64 \times (h - 1)) = H^h(f), f \in \{1, 2, \dots, 64\}, h \in \{1, 2, \dots, 6\} \quad (12)$$

where  $fv$  is the features extracted by using the frustum pattern.

#### 4. The Proposed Learning Model: Frustum154

The main objective of the proposed model is to achieve excellent classification ability with biomedical signals for classification problems. The model comprises a feed-forward network and a hand-modeled architecture. The used architecture has feature extraction, feature selection, and classification phases. A machine learning model is proposed for feature extraction, comprising a multilevel method. Effective wavelet decomposition (TQWT) is utilized as a decomposition method. By using TQWT, 153 subbands are generated. Two feature extraction functions are used to generate fused features, i.e., a statistical generator and frustum pattern. Using these functions, 154 feature vectors (a raw sEMG signal and 153 subbands) are generated. Misclassification rates of these vectors are calculated using kNN and SVM classifiers (herein, kNN and SVM are utilized as loss functions) and a loss array is created. The top 20 feature vectors are selected using loss values, which are merged to create the final feature vector. INCA is applied to automatically choose the most discriminative features. In the classification phase, kNN or SVM are used to demonstrate the excellent classification ability of the created features. A graphical summary of the proposed model is shown in Figure 3.



**Figure 3.** Graphical summary of Frustum154. In the first step, TQWT with multiple parameters is applied to a sEMG signal and 153 subbands (SBs) are calculated. Then, 154 feature vectors (153 subbands + sEMG) are created by applying the proposed frustum pattern and statistical feature extractor. The Frustum pattern generates 384 features, while 30 features are created using statistics. Therefore, the length of each feature vector is computed as 414. By deploying a shallow classifier with 10-fold cross-validation, misclassification rates (loss values) are calculated and the top 20 features are selected according to the loss values. These top features are merged and a feature vector comprising  $414 \times 20 = 8280$  features is obtained, from which INCA chooses the most informative ones, which these are classified using kNN or SVM with 10-fold cross-validation.

Figure 3 describes the proposed model. Pseudocode of the presented model is shown in Algorithm 1, and the transition table of this learning model is given in Table 1.

---

**Algorithm 1:** Pseudocode of the introduced Frustum154.

---

Input: sEMG dataset

Output: Results.

---

00: Read each sEMG signal from datasets.

01: Merge the channels

02: Apply for TQWT decomposition to calculate 153 subbands.

03: Extract statistical and texture features from sEMG signal and subbands.

04: Obtain a 154-feature vector with a length of 414 (=384 + 30).

05: Calculate the misclassification rate of each feature vector.

06: Choose the best 20 feature vectors using the calculated misclassification rates.

07: Merge the 20 feature vectors selected to create final feature vector.

08: Choose the most informative feature by applying INCA.

09: Classify the chosen features using kNN or SVM with 10-fold cross-validation.

---

**Table 1.** Transition table of the presented Frustum154.

Operation	Parameter	Output
Channel merging	Two channels	The used datasets comprise two-channeled sEMG signals. These channels are concatenated to use both channels.
TQWT	Q = 1, 2, 3, 4 r = 2, 4, 6, 8 J = 6, 24, 46, 73	153 subbands
Frustum pattern	Forty-nine overlapping blocks are used. The kernel function is ternary and the threshold value is chosen as half of the standard deviation of the signal	154 feature vectors with a length of 384
Statistical feature extraction	We applied well-known statistical moments	154 feature vectors with a length of 30
Feature merging		154 feature vectors with a length of 414
Normalization	Min-max normalization	154 feature vectors are normalized
Loss value generation	Cubic SVM and kNN (1NN with L1-norm) with 10-fold cross-validation. Herein, greedy model has been used. For DB1 (first database), kNN is the best classifier. For others (DB2 and DB3), the best loss value generator is Cubic SVM.	154 loss values
Top 20 feature vectors selection	Loss array	20 feature vectors
Feature merging	Concatenation function	Final feature vector with a length of 8280

---

Table 1. Cont.

Operation	Parameter	Output
INCA selector	Iteration range: [100, 512] Classifier: SVM or kNN (greedy search-based)	Length of the chosen feature vectors DB1: 279 DB2: 277 DB3: 295
Classification	kNN: k is 1, distance is Manhattan and voting is none. Validation is 10-fold CV. SVM: Kernel scale is auto, kernel is 3rd degree polynomial, C value is 1 and coding is one-vs-one. Validation is 10-fold CV.	Predicted values

More explanations of the FrustumNet41 are given in subsections.

#### 4.1. Feature Generation

The most complex and first phase of the Frustum154 is feature extraction. In this paper, a machine learning method is proposed as a feature extraction method, since this phase uses feature creation and classification methods together to generate the most appropriate features. The concatenated sEMG signal (channels 1 and 2 were merged to obtain the sEMG signal) and TQWT subbands (153 subbands were generated) were utilized for feature generation. By deploying this feature extractor (Frustum Pattern and Statistics), textural and statistical features were generated from the 154 signals (raw sEMG signal and 153 TQWT subbands). Therefore, 154 feature vectors were created, which were then merged to create the final feature vector. To better explain the presented feature generation method, the steps of this process are given in below.

**Step 0:** Read sEMG signals and concatenate channels to obtain the input sEMG signal.

$$sEMG = conc(Ch_1, Ch_2) \quad (13)$$

Herein, the concatenated sEMG describes the sEMG signal,  $Ch_1$  and  $Ch_2$  are the first and second channels of the sEMG signal and  $conc(\cdot)$  is the concatenation function.

**Step 1:** Decompose sEMG using the TQWT decomposition model. We used multiple parameters TQWT.

$$SB1 = TQWT(sEMG, 1, 2, 6) \quad (14)$$

$$SB2 = TQWT(sEMG, 2, 4, 24) \quad (15)$$

$$SB3 = TQWT(sEMG, 3, 6, 46) \quad (16)$$

$$SB4 = TQWT(sEMG, 4, 8, 73) \quad (17)$$

$$SB = [SB1, SB2, SB3, SB4] \quad (18)$$

where  $SB1, SB2, SB3, SB4$  are generated wavelet coefficients (subbands) by applying TQWT with four variable parameters. By applying these parameters-based TQWT ( $TQWT(\cdot)$ ), 7, 25, 47, and 74 subbands are created. These subbands are collected in a structure ( $SB$ ) which contains 153 subbands. This step is parametric. Variable parameters can be used in this phase.

**Step 2:** Extract features by deploying the proposed Frustum pattern and statistical features.

$$fev^1 = conc(FP(sEMG), SE(sEMG)) \quad (19)$$

$$fev^{j+1} = conc(FP(SB^j), SE(SB^j)), j \in \{1, 2, \dots, 153\} \quad (20)$$

where  $fev$  are feature vectors and this model creates 154 feature vectors,  $FP(\cdot)$  is frustum pattern and  $SE(\cdot)$  is the statistical extractor. We used 15 statistical moments in the used statistical feature extractor; the used moments are tabulated in Table 2.

**Table 2.** The used statistics for feature extraction.

No.	Statistics	No.	Statistics	No.	Statistics
1	Mean	6	Maximum	11	Kurtosis
2	Median	7	Minimum	12	Skewness
3	Variance	8	Standard deviation	13	Higuchi
4	Shannon entropy	9	Range	14	Energy
5	Log entropy	10	Sure entropy	15	Root mean square error

These statistical moments (see Table 2) were applied to the raw signal and absolute values of the signal.

Thirty statistical features were generated by applying these statistical moments. In this respect,  $FP(\cdot)$  generated 384 and  $SE(\cdot)$  extracted 30 features from each subband/sEMG signal. By merging these features, 414 (=384 + 30) features were generated from each input (subband/sEMG).

**Step 3:** Normalize features using min–max normalization.

$$fev^k = \frac{fev^k - \min(fe v^k)}{\max(fe v^k) - \min(fe v^k)}, k \in \{1, 2, \dots, 154\} \quad (21)$$

**Step 4:** Apply a loss generation function (kNN or SVM with 10-fold cross-validation) to generate features and calculate the loss array. The main objective of this step is to select the most significant subbands for feature extraction. In order to choose the most significant subbands according to the proposed frustum pattern and statistical feature extraction (see Table 2), loss values had to be calculated. Therefore, shallow classifiers were utilized as the loss value generator.

**Step 5:** Choose the top 20 feature vectors and concatenate them to obtain features with a length of 8280. This architecture is parametric and we selected the top 20 feature vectors to create the final feature vector. A variable number of features or a threshold point can be used to create the final feature vector.

After creating the final feature vector, the optimal number of features was chosen using INCA selector; details are presented in Section 4.2.

#### 4.2. Feature Selection

Herein, an iterative feature function, i.e., INCA, was used to select the best feature combination. This is an iterative version of the NCA classifier; its primary purpose is to solve automated optimal feature vector selection using NCA, since NCA cannot select the best number of features without using the trial and error method. INCA was presented by Tuncer et al. [34] in 2020, and is a very effective feature selector. The parameters of the applied INCA are defined in Table 1. For cubic SVM and kNN (1NN with L1-norm) with 10-fold cross-validation were used. For DB1 (first database), kNN is the best classifier. For others (DB2 and DB3), the best loss value generator is Cubic SVM. This generator chooses 413 features and selects the best combination according to misclassification rates. In this work, three datasets were used for testing. INCA chose 279, 277, and 295 features for the DB1, DB2, and DB3 datasets, respectively. The steps for the NCA are given below.

**Step 6:** Deploy INCA to select the most informative features.

#### 4.3. Classification

kNN and SVM classifiers were used and results were obtained. Therefore, the most appropriate classifier was selected to solve the problem. We used two classifiers and the proposed Frustum154 chose the most effective one. The properties of the used classifiers are tabulated in Table 1.

**Step 7:** We then calculated the results using the kNN or SVM classifier. The hyperparameters of the used classifiers are as follows. For the kNN classifiers, k is 1, distance is Manhattan and voting is none. The hyperparameters of the SVM classifier are as follows: Kernel scale is auto, kernel is 3rd degree polynomial, C (box constraint) value is 1 and coding is one-vs-one. Moreover, ten-fold CV was used to validate these classifiers.

## 5. Experimental Protocol

### 5.1. Experimental Setup

In this paper, three publicly-available sEMG signal datasets were used to evaluate Frustum a pattern-based classification model. The presented model is self-organized. To implement it, MATLAB 2021b was used. We programmed this model using the following functions: main, TQWT, Frustum\_Pattern, statistics, loss\_calculator, feature\_vector\_selector, INCA and classification. This model was implemented on a personal computer (PC) with a simple configuration, as it is lightweight and there is no need to use any unusual hardware.

### 5.2. Validation

The presented model is a classification model. In the classification and loss value generation phases, 10-fold cross-validation was used to obtain robust results. In this validation technique (10-fold cross-validation), the observations were divided randomly into 10 folds, and the average value of the results was calculated.

### 5.3. Results

To evaluate the presented Frustum154, three sEMG signals datasets were used for the general classification results. We used recall, precision, F1-score, and accuracy performance metrics to obtain measurements. The present model can select the best classifier according to the problem. The results of the DB1 were calculated by utilizing the kNN classifier. SVM was used for the other two datasets (DB2 and DB3). Furthermore, 10-fold cross-validation was utilized as a validation model to obtain robust classification results. The calculated confusion matrices are tabulated in Table 3.

**Table 3.** The confusion matrix of the DB1.

True Label	Predicted Label					
	C	H	L	P	S	T
C	150	0	0	0	0	0
H	0	149	0	1	0	0
L	0	0	147	1	0	2
P	0	0	1	149	0	0
S	0	0	0	0	150	0
T	1	1	1	2	0	145
Recall (%)	100	99.33	98	99.33	100	96.67
Precision (%)	99.34	99.33	98.66	97.39	100	98.64
F1 (%)	99.67	99.33	98.33	98.35	100	97.64

Table 3 shows that the proposed Frustum154 achieved 100% class-wise accuracies (recall) for cylindrical and spherical movements, as well as 100% F1-score for spherical movement. The confusion matrix for DB2 is tabulated in Table 4.

**Table 4.** The confusion matrix of the DB2.

True Label	Predicted Label					
	C	H	L	P	S	T
C	297	1	0	0	2	0
H	3	291	0	2	0	4
L	1	1	273	18	0	7
P	0	2	19	273	0	6
S	1	0	0	0	299	0
T	0	1	11	12	0	276
Recall (%)	99	97	91	91	99.67	92
Precision (%)	98.34	98.31	90.10	89.51	99.34	94.20
F1 (%)	98.67	97.65	90.55	90.25	99.50	93.09

For DB2, the best class was spherical movement (S), for which Frustum154 achieved 99.67% recall. The worst categories were Palmar and Lateral, which reached 91% classification accuracies.

The last signal dataset was DB3; this is a merged dataset. Table 5 shows the confusion matrix for DB3.

**Table 5.** The confusion matrix of the DB3.

True Label	Predicted Label					
	C	H	L	P	S	T
C	442	5	0	0	3	0
H	4	431	0	7	2	6
L	0	0	421	19	0	10
P	0	2	14	425	0	9
S	5	1	0	0	444	0
T	2	6	18	14	0	410
Recall (%)	98.22	95.78	93.56	94.44	98.67	91.11
Precision (%)	97.57	96.85	92.94	91.40	98.89	94.25
F1 (%)	97.90	96.31	93.24	92.90	98.78	92.66

Spherical movement was the best class for DB3, as was the case with DB1 and DB2. The worst category was Tip, with Frustum154 achieving 91.11% recall.

Based on Tables 3–5, the overall results are tabulated in Table 6.

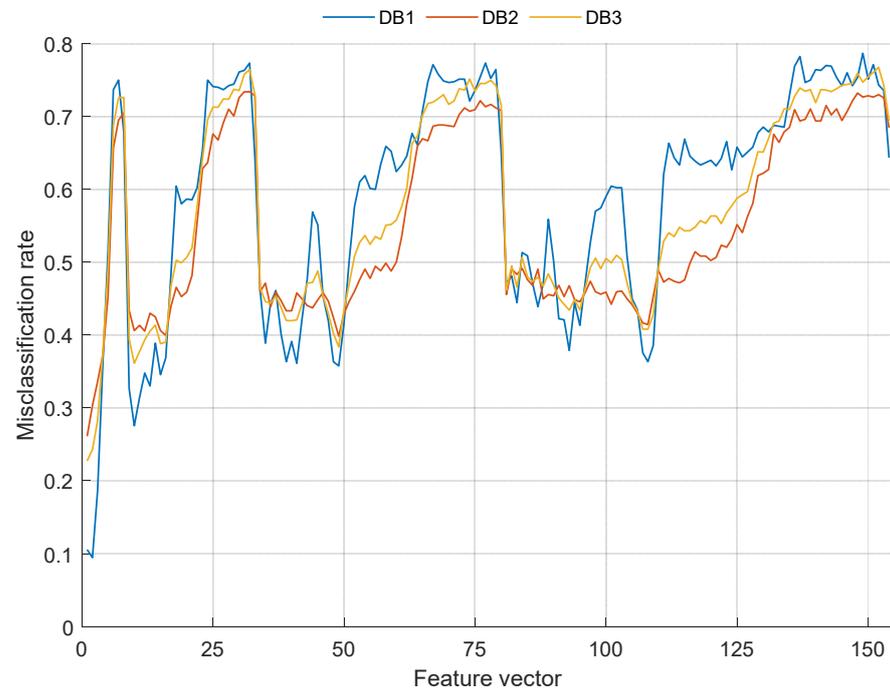
**Table 6.** Overall results (%) of the proposed Frustum154 for the used datasets.

Performance Metrics	DB1	DB2	DB3
Accuracy	98.89	94.94	95.30
Precision	98.89	94.97	95.32
F1	98.89	94.95	95.30

Table 6 shows that Frustum154 achieved 98.89%, 94.94% and 95.30% classification accuracies for DB1, DB2 and DB3, respectively.

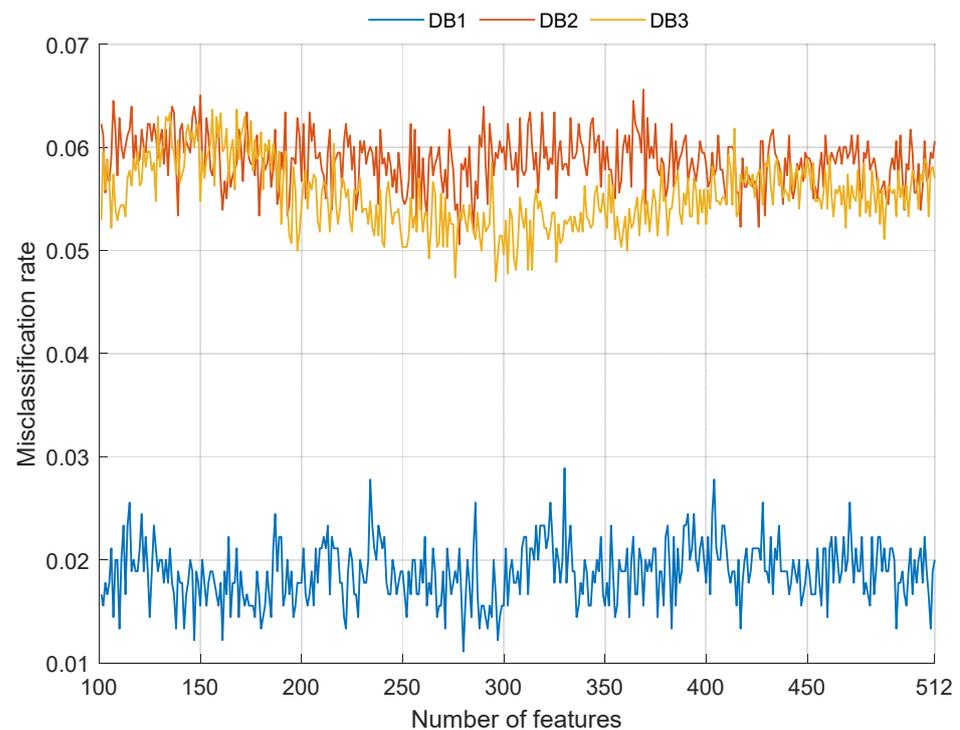
Moreover, the differences among the used feature selection methods (we used two feature selection approximations, i.e., loss value-based selection in the feature extraction

and the INCA model) are explained below. To choose the most appropriate feature vectors, loss values were calculated; these error rates are shown in Figure 4.



**Figure 4.** The calculated misclassification rates.

As shown in Figure 4, the best accuracies of the individual feature vector for DB1, DB2, and DB3 were calculated as 90.56%, 73.89%, and 77.30% respectively. Feature merging (top 20 features concatenation) and INCA were applied to increase these classification accuracies. The INCA feature selection process is shown in Figure 5.



**Figure 5.** INCA feature selection process. Misclassification rates according to the number of features.

The feature merging and INCA processes increased the accuracy rates from 90.56%, 73.89%, and 77.30% to 98.89%, 94.94%, and 95.30% for DB1, DB2, and DB3 respectively.

#### 5.4. Time Complexity Analysis

A time complexity analysis was conducted. Big theta notation was used to present the general results. By using this notation, the training and testing complexities of our presented Frustum154 could be computed; see Table 7.

**Table 7.** Time complexity computation of the proposed Frustum154.

Phase	Step	Training	Test
Feature extraction	Feature vectors creation using TQWT and frustum pattern	$\theta(tndlognd)$	$\theta(tnlogn)$
	Feature vector selection using loss values	$\theta(tdfk)$	$\theta(1)$
	Feature concatenation	$\theta(fd)$	$\theta(1)$
Feature selection	INCA	$\theta(fdkm)$	$\theta(h)$
Classification	kNN/SVM	$\theta(fdk)$	$\theta(k)$
	Total	$\theta(tndlognd + tdfk + fdkm)$	$\theta(tnlogn + h + k)$

Here, variables were used to calculate asymptotic notation as follows.  $t$  is the number of subbands,  $n$  defines the length of the sEMG signal,  $d$  is the number of observations,  $k$  defines the time complexity coefficient of the parameter,  $f$  is the number of features and  $m$  is the number of iterations in INCA. As shown in Table 7, the time complexity of this model is linear.

## 6. Discussion

This research presents a new classification network to detect six basic hand movements using sEMG signals. The proposed model, Frustum154, creates 154 feature vectors and selects the 20 most appropriate feature ones to create final features. The results (see Section 5) clearly demonstrate the success of the presented feature generation network and a new attribute, i.e., shoelaces, was investigated. To better illustrate the success of the model, a performance comparison was performed; the results are tabulated in Table 8.

**Table 8.** Results (%) of prior sEMG signal classification methods and those of Frustum154.

Study	Method	Dataset	Accuracy (%)
Subasi and Qaisar [41]	Statistical feature extraction	DB1	94.11
Nishad et al. [17]	Statistical (entropy) feature extraction with TQWT decomposition	DB1	98.55
Iqbal et al. [42]	Singular value decomposition and principal component analysis (SVD+PCA) and kNN classifier	DB1	86.71
Sapsanis et al. [43]	Statistics and empirical mode decomposition (EMD) transformation	DB1	86.64
Coskun et al. [19]	One dimensional convolutional neural network (1D-CNN)	DB1	94.94
Tsinganos et al. [44]	Convolutional neural network	DB1	72.06

Table 8. Cont.

Study	Method	Dataset	Accuracy (%)
Rabin et al. [16] method	Short time Fourier transform-based feature generation and principle component analysis/diffusion map-based feature reduction + kNN	DB1	76.4
		DB1	98.89
		DB2	94.94
		DB3	95.30

To the best of our knowledge, to date, no method has utilized DB2 and DB3. Therefore, we cannot perform any comparisons of such cases. We obtained over 94% classification accuracies for all datasets, indicating the high classification rates of our model. Comparisons were made using DB1, since this is the simplest dataset. Subasi and Qaisar [41] presented a statistical feature extraction-based model using DB1 which achieved 94.11% accuracy. They showed the classification ability of the statistical features but their classification result was not good. Nishad et al. [17] presented a TQWT and statistical feature extraction-based model. They used DB1 to calculate the results. DB1 contains five subjects, and the authors calculated the results for each. To calculate the overall performance, they used the average values. They did not use all of the dataset to get general results; other models [42,43] have used the same strategy. Coskun et al. [19] presented a one-dimensional, CNN-based deep learning model and achieved 94.94% accuracy. Tsinganos et al. [44] introduced a CNN-based deep learning model but did not achieve satisfactory classification results. Deep learning models have high computational complexity, since many parameters need to be optimized. A self-organized hybrid, hand-crafted feature-based model is presented in this research. To show the classification ability of the presented Frustum pattern-based model, three sEMG datasets for basic hand-movements were used; our model yielded excellent classification results. It is worth noting that:

- The feature generation capabilities of the frustum pattern/graph were investigated and the sEMG classification was found to be highly successful;
- To maximize the effectiveness of TQWT, multiple parameter-based TQWT was used and 153 subbands were generated;
- An improved feature selector (INCA) was used;
- A novel hand-modeled learning method is proposed;
- The proposed Frustum154 achieved a higher classification rate than deep learning models (see Table 8);
- The present model showed good general classification success.

The proposed model could also be applied to more complex and larger datasets; this will be explored in future work.

## 7. Conclusions

The objective of a machine learning model is to achieve excellent classification with low execution time; however, there is usually a tradeoff. To overcome this problem, a self-organized model has been presented and three sEMG signal datasets have been used to depict the general efficacy of the presented model. A novel, hand-modeled feature selection-based, basic hand-movement classification network using a multilevel feature generation method is presented. This approach was inspired by deep feature networks and it has low- and high-level feature generation capabilities. The proposed Frustum154 method generates 154 features, selects the top 20 feature vectors, and chooses the most discriminative ones by deploying the INCA selector. FrustumNet154 has been tested on three datasets, achieving good performance with all three. The accuracies for these datasets were 98.89%, 94.94% and 95.30%, respectively. In the literature (see Table 8), the first sEMG dataset was used

to calculate the classification results. This model solved this problem and all of them were used to test the performance of Frustum154. Most prior models calculate results for each subject and then use the average to achieve good classification ability. In contrast, Frustum154 calculates the results using all of the sEMG observations, allowing it to achieve superior classification rates. This research used two channeled sEMG signals. Therefore, this model could be employed with low-cost exoskeleton prosthetic hands (EPH) or smart gloves. Based on our findings, the proposed approach is a successful classification model for one-dimensional signals (sEMGs). Our approach motivates new low-cost and smart EPHs and smart gloves which could be used in physiotherapy and orthopedics clinics. New smart sEMG signal monitoring applications can be derived by applying our presented model. Other one-dimensional signals can also be classified by applying this model.

**Author Contributions:** Conceptualization, M.B., P.D.B., S.D., T.T., S.K., K.H.C. and U.R.A.; methodology, M.B., P.D.B., S.D., T.T. and K.H.C.; software, T.T.; validation, M.B., P.D.B. and S.K.; formal analysis, M.B. and P.D.B.; investigation, M.B., P.D.B., S.D., T.T., S.K., K.H.C. and U.R.A.; resources, M.B. and P.D.B.; data curation, M.B., P.D.B., S.D. and T.T.; writing—original draft preparation, M.B., P.D.B., S.D. and T.T.; writing—review and editing, M.B., P.D.B., S.D., T.T., S.K., K.H.C. and U.R.A.; visualization, M.B. and P.D.B.; supervision, K.H.C. and U.R.A.; project administration, K.H.C. and U.R.A. All authors have read and agreed to the published version of the manuscript.

**Funding:** This project was partially funded by the Singapore University of Technology and Design (SUTD) Start-up Research Grant (SRG SCI 2019 142).

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** The used dataset was downloaded from <https://archive.ics.uci.edu/ml/datasets/sEMG+for+Basic+Hand+movements>, accessed on 20 January 2022. We thank UCI Machine Learning Repository.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Hargrove, L.J.; Englehart, K.; Hudgins, B. A comparison of surface and intramuscular myoelectric signal classification. *IEEE Trans. Biomed. Eng.* **2007**, *54*, 847–853. [[CrossRef](#)]
2. Kamavuako, E.N.; Rosenvang, J.C.; Horup, R.; Jensen, W.; Farina, D.; Englehart, K.B. Surface versus untargeted intramuscular EMG based classification of simultaneous and dynamically changing movements. *IEEE Trans. Neural Syst. Rehabil. Eng.* **2013**, *21*, 992–998. [[CrossRef](#)] [[PubMed](#)]
3. Kamavuako, E.N.; Scheme, E.J.; Englehart, K.B. Combined surface and intramuscular EMG for improved real-time myoelectric control performance. *Biomed. Signal. Process. Control* **2014**, *10*, 102–107. [[CrossRef](#)]
4. Tuncer, T.; Dogan, S.; Subasi, A. Surface EMG signal classification using ternary pattern and discrete wavelet transform based feature extraction for hand movement recognition. *Biomed. Signal. Process. Control* **2020**, *58*, 101872. [[CrossRef](#)]
5. Yavuz, E.; Eyupoglu, C. A cepstrum analysis-based classification method for hand movement surface EMG signals. *Med. Biol. Eng. Comput.* **2019**, *57*, 2179–2201. [[CrossRef](#)] [[PubMed](#)]
6. Phinyomark, A.; Phukpattaranont, P.; Limsakul, C. Feature reduction and selection for EMG signal classification. *Expert Syst. Appl.* **2012**, *39*, 7420–7431. [[CrossRef](#)]
7. Gokgoz, E.; Subasi, A. Comparison of decision tree algorithms for EMG signal classification using DWT. *Biomed. Signal. Process. Control* **2015**, *18*, 138–144. [[CrossRef](#)]
8. Fang, Y.; Zhou, D.; Li, K.; Ju, Z.; Liu, H. Attribute-driven granular model for EMG-based pinch and fingertip force grand recognition. *IEEE Trans. Cybern.* **2019**, *51*, 789–800. [[CrossRef](#)]
9. Chen, H.; Zhang, Y.; Li, G.; Fang, Y.; Liu, H. Surface electromyography feature extraction via convolutional neural network. *Int. J. Mach. Learn. Cybern.* **2020**, *11*, 185–196. [[CrossRef](#)]
10. Fang, Y.; Zhang, X.; Zhou, D.; Liu, H. Improve inter-day hand gesture recognition via convolutional neural network based feature fusion. *Int. J. Hum. Robot.* **2020**, *18*, 2050025. [[CrossRef](#)]
11. 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](#)]
12. Karabulut, D.; Ortes, F.; Arslan, Y.Z.; Adli, M.A. Comparative evaluation of EMG signal features for myoelectric controlled human arm prosthetics. *Biocybern. Biomed. Eng.* **2017**, *37*, 326–335. [[CrossRef](#)]

13. Arunraj, M.; Srinivasan, A.; Arjunan, S. A real-time capable linear time classifier scheme for anticipated hand movements recognition from Amputee subjects using surface EMG signals. *IRBM* **2020**, *42*, 277–293. [[CrossRef](#)]
14. Hooda, N.; Das, R.; Kumar, N. Fusion of EEG and EMG signals for classification of unilateral foot movements. *Biomed. Signal. Process. Control* **2020**, *60*, 101990. [[CrossRef](#)]
15. Bi, L.; Guan, C. A review on EMG-based motor intention prediction of continuous human upper limb motion for human-robot collaboration. *Biomed. Signal. Process. Control* **2019**, *51*, 113–127. [[CrossRef](#)]
16. Rabin, N.; Kahlon, M.; Malayev, S.; Ratnovsky, A. Classification of human hand movements based on EMG signals using nonlinear dimensionality reduction and data fusion techniques. *Expert Syst. Appl.* **2020**, *149*, 113281. [[CrossRef](#)]
17. Nishad, A.; Upadhyay, A.; Pachori, R.B.; Acharya, U.R. Automated classification of hand movements using tunable-Q wavelet transform based filter-bank with surface electromyogram signals. *Future Gener. Comput. Syst.* **2019**, *93*, 96–110. [[CrossRef](#)]
18. Zhou, D.; Fang, Y.; Botzheim, J.; Kubota, N.; Liu, H. Bacterial memetic algorithm based feature selection for surface EMG based hand motion recognition in long-term use. In Proceedings of the 2016 IEEE Symposium Series on Computational Intelligence (SSCI), Athens, Greece, 6–9 December 2016; pp. 1–7.
19. Coskun, M.; Yildirim, O.; Demir, Y.; Acharya, U.R. Efficient deep neural network model for classification of grasp types using sEMG signals. *J. Ambient. Intell. Humaniz. Comput.* **2021**, 1–14. [[CrossRef](#)]
20. Ouyang, G.; Zhu, X.; Ju, Z.; Liu, H. Dynamical characteristics of surface EMG signals of hand grasps via recurrence plot. *IEEE J. Biomed. Health Inform.* **2013**, *18*, 257–265. [[CrossRef](#)]
21. Ma, R.; Zhang, L.; Li, G.; Jiang, D.; Xu, S.; Chen, D. Grasping force prediction based on sEMG signals. *Alex. Eng. J.* **2020**, *59*, 1135–1147. [[CrossRef](#)]
22. Menon, R.; Di Caterina, G.; Lakany, H.; Petropoulakis, L.; Conway, B.A.; Soraghan, J.J. Study on interaction between temporal and spatial information in classification of EMG signals for myoelectric prostheses. *IEEE Trans. Neural Syst. Rehabil. Eng.* **2017**, *25*, 1832–1842. [[CrossRef](#)] [[PubMed](#)]
23. Khushaba, R.N.; Takruri, M.; Miro, J.V.; Kodagoda, S. Towards limb position invariant myoelectric pattern recognition using time-dependent spectral features. *Neural Netw.* **2014**, *55*, 42–58. [[CrossRef](#)] [[PubMed](#)]
24. Waris, A.; Niazi, I.K.; Jamil, M.; Englehart, K.; Jensen, W.; Kamavuako, E.N. Multiday evaluation of techniques for EMG-based classification of hand motions. *IEEE J. Biomed. Health Inform.* **2018**, *23*, 1526–1534. [[CrossRef](#)]
25. Chada, S.; Taran, S.; Bajaj, V. An efficient approach for physical actions classification using surface EMG signals. *Health Inf. Syst.* **2020**, *8*, 3. [[CrossRef](#)]
26. Wang, Y.; Wu, Q.; Dey, N.; Fong, S.; Ashour, A.S. Deep back propagation–long short-term memory network based upper-limb sEMG signal classification for automated rehabilitation. *Biocybern. Biomed. Eng.* **2020**, *40*, 987–1001. [[CrossRef](#)]
27. Arteaga, M.V.; Castiblanco, J.C.; Mondragon, I.F.; Colorado, J.D.; Alvarado-Rojas, C. EMG-driven hand model based on the classification of individual finger movements. *Biomed. Signal. Process. Control* **2020**, *58*, 101834. [[CrossRef](#)]
28. Pancholi, S.; Joshi, A.M. Electromyography-based hand gesture recognition system for upper limb amputees. *IEEE Sens. Lett.* **2019**, *3*, 1–4. [[CrossRef](#)]
29. Jia, G.; Lam, H.-K.; Liao, J.; Wang, R. Classification of Electromyographic Hand Gesture Signals using Machine Learning Techniques. *Neurocomputing* **2020**, *401*, 236–248. [[CrossRef](#)]
30. Simão, M.; Neto, P.; Gibaru, O. EMG-based online classification of gestures with recurrent neural networks. *Pattern Recognit. Lett.* **2019**, *128*, 45–51. [[CrossRef](#)]
31. Simão, M.; Neto, P.; Gibaru, O. Uc2018 Dualmyo Hand Gesture Dataset. 2018. Available online: [https://zenodo.org/record/1320922#\\_YhR8\\_ZYRVEY](https://zenodo.org/record/1320922#_YhR8_ZYRVEY) (accessed on 7 December 2021).
32. Pizzolato, S.; Tagliapietra, L.; Cognolato, M.; Reggiani, M.; Müller, H.; Atzori, M. Comparison of six electromyography acquisition setups on hand movement classification tasks. *PLoS ONE* **2017**, *12*, e0186132. [[CrossRef](#)]
33. Selesnick, I.W. Wavelet transform with tunable Q-factor. *IEEE Trans. Signal Process.* **2011**, *59*, 3560–3575. [[CrossRef](#)]
34. Tuncer, T.; Dogan, S.; Özyurt, F.; Belhaouari, S.B.; Bensmail, H. Novel Multi Center and Threshold Ternary Pattern Based Method for Disease Detection Method Using Voice. *IEEE Access* **2020**, *8*, 84532–84540. [[CrossRef](#)]
35. Vapnik, V. The support vector method of function estimation. In *Nonlinear Modeling*; Springer: Boston, MA, USA, 1998; pp. 55–85.
36. Vapnik, V. *The Nature of Statistical Learning Theory*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2013.
37. Maillo, J.; Ramírez, S.; Triguero, I.; Herrera, F. kNN-IS: An Iterative Spark-based design of the k-Nearest Neighbors classifier for big data. *Knowl.-Based Syst.* **2017**, *117*, 3–15. [[CrossRef](#)]
38. Sapsanis, C.; Tzes, A.; Georgoulas, G. sEMG for Basic Hand Movements Data Set. UCI Machine Learning Repository 2014. Available online: <https://archive.ics.uci.edu/ml/datasets/sEMG+for+Basic+Hand+movements> (accessed on 7 December 2021).
39. Sapsanis, C.; Georgoulas, G.; Tzes, A. EMG based classification of basic hand movements based on time-frequency features. In Proceedings of the 21st Mediterranean Conference on Control and Automation, Chania, Greece, 25–28 June 2013; pp. 716–722.
40. Nishimura, Y.; Murakami, H. Initial shape-finding and modal analyses of cyclic frustum tensegrity modules. *Comput. Methods Appl. Mech. Eng.* **2001**, *190*, 5795–5818. [[CrossRef](#)]
41. Subasi, A.; Qaisar, S.M. Surface EMG signal classification using TQWT, Bagging and Boosting for hand movement recognition. *J. Ambient. Intell. Humaniz. Comput.* **2020**, 1–16. [[CrossRef](#)]

42. Iqbal, O.; Fattah, S.A.; Zahin, S. Hand movement recognition based on singular value decomposition of surface EMG signal. In Proceedings of the Humanitarian Technology Conference (R10-HTC), 2017 IEEE Region 10, Dhaka, Bangladesh, 21–23 December 2017; pp. 837–842.
43. Sapsanis, C.; Georgoulas, G.; Tzes, A.; Lymberopoulos, D. Improving EMG based classification of basic hand movements using EMD. In Proceedings of the 2013 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), Osaka, Japan, 3–7 July 2013; pp. 5754–5757.
44. Tsinganos, P.; Cornelis, B.; Cornelis, J.; Jansen, B.; Skodras, A. Deep Learning in EMG-based Gesture Recognition. In Proceedings of the 5th International Conference on Physiological Computing System (PhyCS 2018), Seville, Spain, 19–20 September 2018; pp. 107–114.