CNN Confidence Estimation for Rejection-based Hand Gesture Classification in Myoelectric Control

Tianzhe Bao, Syed Ali Raza Zaidi, *Member, IEEE*, Shane Q. Xie, *Senior Member, IEEE*, Pengfei Yang, *Member, IEEE*, and Zhi-Qiang Zhang, *Member, IEEE*

***Abstract*—Convolutional neural networks (CNNs) have been widely utilized to identify hand gestures from surface electromyography (sEMG) signals. However, due to the nonstationary characteristics of sEMG, the classification accuracy usually degrades significantly in the daily living environment involving complex hand movements. To further improve the reliability of a classifier, unconfident classifications are expected to be identified and rejected. In this study, we propose a novel approach to estimate the probability of correctness for each classification. Specifically, a confidence estimation model is established to generate confidence scores (ConfScore) based on posterior probabilities of CNN, and an objective function is designed to train the parameters of this model. In addition, a comprehensive metric that combines the true acceptance rate and the true rejection rate is proposed to evaluate the rejection performance of ConfScore, so that the trade-off between system security and control lag could be fully considered. The effectiveness of ConfScore is verified using data from public databases and our online platform. The experimental results illustrate that ConfScore can better reflect the correctness of CNN classifications than traditional confidence features, i.e., maximum posterior probability and entropy of the probability vector. Moreover, the rejection performance is observed to be less sensitive to variations in rejection thresholds.**

***Index Terms*—CNN, sEMG, Hand Gesture Classification, Model Confidence, Rejection Strategy.**

I. Introduction

SURFACE electromyography (sEMG) is the electrical manifestation of neuro-muscular activities collected by surface electrodes [1]. Thus far, sEMG signals have been widely investigated for upper-limb myoelectric control [2]–[4], and pattern recognition (PR) approaches have been used extensively to identify hand or wrist gestures. Various methods focusing on signal decomposition [5], feature extraction [6], dimension reduction [7], channel optimization [8], classifier design [9], and post-processing [10], etc., have been proposed to improve recognition accuracy.

Although promising accuracy has been achieved in laboratory settings, the accuracy tends to degrade significantly when hand movements are performed in the daily living environment [11] [12] because there are significant variations between training and testing data due to the transient nature of sEMG. These variations can be caused by muscle fatigue, electrode shift, and impedance changes in the electrode-skin interface [12]–[15], etc. However, for most PR approaches such as linear discriminant analysis (LDA) or artificial neural network (ANN), the classifier could only output one of the predefined gestures even though sEMG inputs have varied dramatically from training samples [16]. This issue results in large uncertainties in classification and can cause meaningless or unwanted outcomes. Therefore, the reliability of PR-based myoelectric control is still very limited.

To enhance the PR approaches in prosthetic control using sEMG, confidence estimations are now being investigated for quantitative evaluation of classification uncertainties. A rejection process can be cascaded with the classifier to refuse unconfident classification results, thus improving the reliability of myoelectric control systems by reducing erroneous movements. Numerous confidence estimation methods have been proposed in the past decade. For instance, Scheme et al. [17] calculated the maximum posterior probability of LDA as the confidence metric. Estimated hand motions would be reverted to no movement when the associated confidence was below a given rejection threshold. Sebastian et al. [18] combined the maximum posterior probability of LDA and the root mean square (RMS) of sEMG as confidence features, based on which a cascaded ANN was trained to detect potentially erroneous decisions of LDA. Scheme et al. [19] examined the confidence characteristics of several conventional classifiers, whilst Robertson et al. [16] verified the optimal rejection threshold for myoelectric control driven by support vector machine (SVM). However, these studies exploited mainly the confidence characteristics of traditional machine learning (ML) methods, where both the classification accuracy and rejection performances depend heavily on the design/selection of hand-crafted features (feature engineering); therefore, it is more desirable to develop learning algorithms that can extract representative features from raw data [20].

Currently, deep learning (DL) techniques, particularly convolutional neural networks (CNNs), are becoming popular in hand gesture recognition due to their strong capability of deriving data-dependent features automatically from sEMG [1], and better performance of CNN over traditional ML methods has been reported in previous studies [21]–[26].

This work was supported in part by Engineering and Physical Sciences Research Council (EPSRC) under Grant EP/S019219/1, in part by the School of Electronic and Electrical Engineering, University of Leeds, in part by the National Natural Science Foundation of China (NSFC) under Grant 61972302 and Grant 61962019, and in part by Key Research and Development Program of Shaanxi (2021ZDLGY07-01) *(Corresponding author: Pengfei Yang; Zhi-Qiang Zhang.)*

Tianzhe Bao, Syed Ali Raza Zaidi, Shane Q. Xie and Zhi-Qiang Zhang are with Institute of Robotics, Autonomous System and Sensing, School of Electronic and Electrical Engineering, University of Leeds, LS2 9JT, UK (e-mail: eltb@leeds.ac.uk; S.A.Zaidi@leeds.ac.uk; S.Q.Xie@leeds.ac.uk; Z.Zhang3@leeds.ac.uk).

Pengfei Yang is with School of Computer Science and Technology, Xidian University, China (e-mail: pfyang@xidian.edu.cn).

Recently, many researchers have started to link the class probability distribution to the confidence of CNN classification accuracy. For example, Ranjan et al. [27] predicted the task labels through the HyperFace network and recognized regions as faces when the maximum probability was above a certain threshold. Zhang et al. [28] utilized the probability distribution of CNN as its confidence feature and designed a decision fusion rule for remotely sensed image classification. Wang et al. [29] proposed the “I Don’t Know“ (IDK) prediction cascades framework leveraging the entropy of CNN likelihoods. Wan et al. [30] designed a Confidence Network (ConfNet) on the basis of probability distribution to generate confidence features and evaluate classification correctness. However, all these studies were conducted for computer vision tasks. To our best knowledge, the CNN confidence estimation and rejection analysis have yet to be investigated in myoelectric control.

In this study, we propose a novel approach to estimate the probability of correctness for each output of the classifier. The main contributions of the paper include: 1) a confidence estimation model established to generate confidence scores (ConfScore) based on posterior probabilities of CNN, and an objective function designed to train parameters on this model; and 2) a comprehensive metric which combines the true acceptance rate and the true rejection rate proposed to evaluate rejection performances so that the trade-off between system security and control lag could be fully considered. Effectiveness of ConfScore was verified using data from public databases and our online platform. Experimental results illustrate that ConfScore can better reflect the correctness of CNN classifications than traditional confidence features, i.e., maximum posterior probability and entropy of the probability vector. Moreover, the rejection performances are observed to be less sensitive to variations of rejection thresholds.

The reminder of the paper is organized as follows. Section II first introduces the framework of the confidence-based rejection for hand gesture recognition. Section II then presents the proposed CNN classifier, the confidence estimation model, the rejection rule, and the comprehensive evaluation metric. Section III introduces the setups of public databases and the online experiment. Section IV demonstrates the experiment results. Section V presents the analysis of ConfScore in both confidence estimation and rejection evaluation. In Section VI, conclusions are drawn, and the future work is presented.

II. Methodology

A. CNN-based Confidence Estimation and Rejection

As illustrated in Fig. 1, when a new classification is made by CNN, the posterior probability vector $\boldsymbol{p} = [p_1, p_2, \cdots p_m]$ is produced in the softmax layer, where m represents the number of hand gestures to be classified. Utilizing this probability vector, the confidence estimation model can generate confidence scores to indicate the probability of correctness for each classification. Based on confidence scores, a threshold-based rejection process can be implemented to decide either to accept the estimated class or to revert the unconfident prediction to a no motion state. This rejection rule can be regarded as a flexible binary classifier cascaded to CNN [29] and has been widely adopted in myoelectric control systems [16], [17], [19] and computer vision tasks [27], [29]–[31].

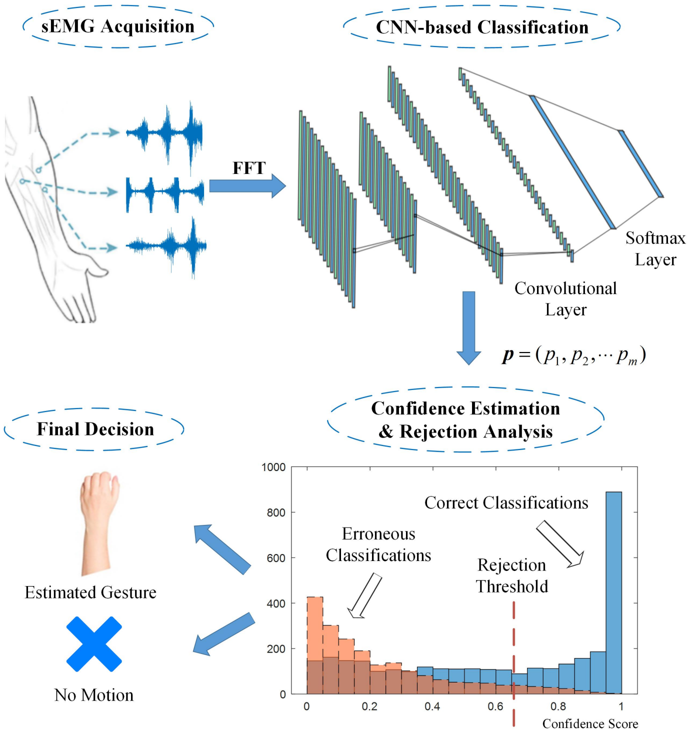

Fig. 1: The framework for rejection-based hand gesture classification using CNN confidence.

B. CNN Classifier

Since CNN is a neural network originally designed for processing data in the form of multiple arrays such as images, we need to construct a matrix $\boldsymbol{X}$ from sEMG signals. Specifically, a sliding window method is utilized to obtain $\boldsymbol{X}$ from a segment of multichannel signals, thus $\boldsymbol{X}$ is designed as a $1 \times L \times C$ matrix, where L denotes the window length and C represents the number of sensor channels. This one-dimensional (1D) multichannel format [32] is utilized since spatial correlations of all sEMG channels can be efficiently exploited via convolution operations. The sEMG matrix $\boldsymbol{X}$ is finally obtained by applying fast Fourier transform (FFT) to signals of each channel, as the spectrum of sEMG is observed to be less noisy and thereby more distinguishable than sEMG data in the temporal domain.

Under myoelectric control $\boldsymbol{X}$ is normally a small-scale input for CNN, thus architectures based on LeNet-5 [33] are still dominant [34]–[36]. In this study, we adopted a single stream CNN for the trade-off between classification accuracy and computational efficiency. As illustrated in Table I, the CNN presented consists of two convolutional layers, one fully connected layer and a softmax layer. After each convolutional layer, a batch normalization layer (for model robustness by reducing covariate shifts in intermediate representations after convolutional operations [37]), a ReLU layer (for nonlinearization), a max-pooling layer (for subsampling) and a dropout layer (for regularization) are attached subsequently.

TABLE I: Layers configuration of implemented CNN

Input: 1 × L × C sEMG matrix after FFT
1D Convolution, 32 kernels in size of 3
Batch Normalization
ReLU
MaxPooling
Dropout
1D Convolution, 64 kernels in size of 3
Batch Normalization
ReLU
MaxPooling, pool size of 3
Dropout
Fully Connected Layer
Softmax Layer

Actually, our CNN classifier can be regarded as a simplification of the network proposed in [36]. We empirically observed that our simplified network can also work efficiently on these public datasets but with less training time.

As mentioned above, CNN can produce a posterior probability vector $\boldsymbol{p} = [p_1, p_2, \cdots p_m]$ for each classification. Herein we denote G as the hand gesture, and p_k ($k = 1, ..., m$ and $\sum_1^m p_k = 1$) corresponds to the probability of the k^{th} gesture $P\left(G_k|\boldsymbol{X}\right)$. The gesture-owning maximum probability is taken as the final prediction:

$$\hat{G} = \arg\max_{G_k} \left\{P\left(G_k|\boldsymbol{X}\right)\right\}_{k=1,...,m} \tag{1}$$

Ideally $\boldsymbol{p}$ is expected to be a one-hot vector for a correct prediction, whilst a uniform distribution is reported when CNN becomes quite uncertain [30]. Thus $\boldsymbol{p}$ can be utilized to exploit confidence features for CNN.

C. Confidence Estimation

To indicate how confident the CNN classifier is about its prediction, a confidence estimation model is proposed by applying a zero-order smooth-step function to the weighted posterior probability distribution of CNN. The mathematical expression of this confidence model is

$$\text{ConfScore}\left(\boldsymbol{p}^*, \boldsymbol{\beta}\right) = \begin{cases} 0 & \boldsymbol{p}^*\boldsymbol{\beta}^T \leq \gamma_1 \\ \dfrac{\boldsymbol{p}^*\boldsymbol{\beta}^T - \gamma_1}{\gamma_2 - \gamma_1} & \gamma_1 < \boldsymbol{p}^*\boldsymbol{\beta}^T < \gamma_2 \\ 1 & \boldsymbol{p}^*\boldsymbol{\beta}^T \geq \gamma_2 \end{cases} . \tag{2}$$

where $\boldsymbol{p}^* = [p_1^*, p_2^*, \cdots p_m^*]$ is obtained by sorting the posterior probability vector $\boldsymbol{p}$ in a descending order. The element p_1^* is the largest posterior probability in $\boldsymbol{p}$ and p_m^* is the smallest. $\boldsymbol{\beta} = [\beta_1, \beta_2, \cdots \beta_m]$ is a coefficient vector, γ_1 and γ_2 are user-defined hyper-parameters to decide left and right edges. Similar to Confnet proposed in [30], $\text{ConfScore}\left(\boldsymbol{p}^*, \boldsymbol{\beta}\right)$ can be regarded as a feed-forward neural network cascaded with the softmax layer of CNN. Due to characteristics of the smooth-step function, outputs of $\text{ConfScore}\left(\boldsymbol{p}^*, \boldsymbol{\beta}\right)$ are mapped between [0, 1]. In the following part, we use ConfScore to denote estimations of $\text{ConfScore}\left(\boldsymbol{p}^*, \boldsymbol{\beta}\right)$.

In this study, $\boldsymbol{\beta}$ is designed as a learnable parameter that can be tuned in a supervised manner. Given a group of classification results $T_r = \left\{\left(\boldsymbol{p}_1^*, \hat{G}_1, \widetilde{G}_1\right), \cdots, \left(\boldsymbol{p}_N^*, \hat{G}_N, \widetilde{G}_N\right)\right\}$ as the training data of $\text{ConfScore}\left(\boldsymbol{p}^*, \boldsymbol{\beta}\right)$, where $\hat{G}_j$ is the estimated class obtained by Eq. (1), $\widetilde{G}_j$ ($j = 1, ..., N$) denotes the ground truth gesture for the j^{th} classification, and we then relabel T_r using l_j by

$$l_j = \begin{cases} 1 & \hat{G}_j = \tilde{G}_j \\ -1 & \hat{G}_j \neq \tilde{G}_j \end{cases} \tag{3}$$

where a relabelled dataset $T = \left\{\left(\boldsymbol{p}_1^*, l_1\right), \cdots, \left(\boldsymbol{p}_N^*, l_N\right)\right\}$ can be obtained to provide ground-truth labels for the supervised learning of $\text{ConfScore}\left(\boldsymbol{p}^*, \boldsymbol{\beta}\right)$. Specifically, the ground truth of $\text{ConfScore}\left(\boldsymbol{p}^*, \boldsymbol{\beta}\right)$ is +1 for correct CNN outcomes or -1 for erroneous ones. To tune parameter $\boldsymbol{\beta}$ effectively, a metric should be determined to evaluate the performance of $\text{ConfScore}\left(\boldsymbol{p}^*, \boldsymbol{\beta}\right)$ in T. Considering that higher scores should be correlated with more accurate predictions, Wan et al. [30] defined the mean effective confidence (MEC) as follows:

$$\text{MEC} = \frac{1}{N}\sum_{j\in T} \text{ConfScore}\ \left(p_j^d, \boldsymbol{\beta}\right) * l_j \tag{4}$$

where $\text{MEC} \in [-1, 1]$. From Eq. (3) and Eq. (4) we can see that a larger MEC can be obtained when correct classifications have the higher ConfScore results whilst erroneous decisions are with the lower scores. In other words, when a better confidence estimation is conducted for T, the separation between correct and erroneous CNN predictions is expected to be more distinguishable [19].

However, as shown in Eq. (4), MEC averages the confidence features of all samples in T. Thus MEC is sensitive to an unbalanced T which is composed of either too many correct or erroneous classifications. To solve this problem, we further define the Balanced MEC (BMEC) as

$$\begin{aligned} \text{BMEC} = & \frac{1}{N_1}\sum_{i\in T_C} \text{ConfScore}\left(\boldsymbol{p}_i^*, \boldsymbol{\beta}\right) * l_i \\ & + \frac{1}{N_2}\sum_{j\in T_E} \text{ConfScore}\left(\boldsymbol{p}_j^*, \boldsymbol{\beta}\right) * l_j \end{aligned} \tag{5}$$

where $\text{BMEC} \in [-1, 1]$, T_C is only composed of correct classifications ($l = 1$) whilst T_E only consists of erroneous classifications ($l = -1$). The number of samples in T_C and T_E is defined as N_1 and N_2, respectively. Compared with MEC, BMEC is more robust to the imbalance of T_C and T_E.

Based on Eq. (5), the optimization of $\boldsymbol{\beta}$ can be defined as

$$\hat{\boldsymbol{\beta}} = \arg\max_{\boldsymbol{\beta}} \text{BMEC} \tag{6}$$

Since Eq. (6) works as the objective function and is non-differentiable, heuristic methods can be adopted to find obtain the local optimal $\hat{\boldsymbol{\beta}}$. Herein, we apply the genetic algorithm (GA) in which solutions evolve efficiently over generations. GA is one of the most widely applied evolution algorithms in the optimization of intricate problems in different fields. Compared with many other heuristic algorithms, it is believed to own better global searching capability [38]. The whole process can be summarized in Algorithm 1.

```
Algorithm 1: The Proposed ConfScore $(\boldsymbol{p}^*, \boldsymbol{\beta})$.
Input: A group of CNN outcomes $T_r$, hyper-parameters $\gamma_1$ and $\gamma_2$.
Output: Optimal weights $\hat{\boldsymbol{\beta}}$
1: Construct the relabelled dataset $T$ from $T_r$.
2: Initialize $\boldsymbol{\beta}$.
3: Use GA to search $\hat{\boldsymbol{\beta}}$:
4:     Based on Eq. (2), calculate ConfScore $(\boldsymbol{p}^*, \boldsymbol{\beta})$ for each sample in $T$.
5:     Based on Eq. (5), calculate BMEC using outputs of the obtained ConfScore $(\boldsymbol{p}^*, \boldsymbol{\beta})$ together with corresponding labels $l$ in $T$.
6:     Update $\boldsymbol{\beta}$ using GA operators, where BMEC works as the objective function.
7: Until GA is converged.
8: Return $\hat{\boldsymbol{\beta}}$ to obtain the final confidence model.
```

D. Rejection Rule

In the rejection process, CNN classifications whose ConfScore is smaller than a user-defined threshold $\alpha \in (0, 1)$ should be rejected to no motion states. Thus, the rejection function can be formulated as

$$\mathrm{R}(\boldsymbol{p}^*, \alpha) = \begin{cases} \text{accept} & \text{ConfScore}\left(\boldsymbol{p}^*, \hat{\boldsymbol{\beta}}\right) \geq \alpha \\ \text{reject} & \text{otherwise} \end{cases}. \quad (7)$$

As illustrated in Fig. 1, once $\hat{\boldsymbol{\beta}}$ is calculated and α is determined, these parameters can be applied in the rejection framework to decrease erroneous movements and thereby enhance model reliability in myoelectric control systems.

E. Rejection Analysis

Since $\mathrm{R}(\boldsymbol{p}^*, \alpha)$ works as a binary classifier to further identify CNN predictions, the rejection results can be divided into true acceptance (TA) cases, false acceptance (FA) cases, false rejection (FR) cases and true rejection (TR) cases. Descriptions of TA/FA/FR/TR are listed in Table II. Based on these cases, the true acceptance rate (TAR) can be calculated to denote the proportion of correct CNN estimations to be accepted by $\mathrm{R}(\boldsymbol{p}^*, \alpha)$, whereas the true rejection rate (TRR) is the rejection ratio of erroneous classifications:

$$\begin{aligned} \mathrm{TAR} &= \frac{\sum \mathrm{TA}}{\sum \mathrm{TA} + \sum \mathrm{FR}} \\ \mathrm{TRR} &= \frac{\sum \mathrm{TR}}{\sum \mathrm{TR} + \sum \mathrm{FA}} \end{aligned} \quad (8)$$

TRR represents the rejection efficiency whilst TAR corresponds to the cost. According to previous work [19], a trade-off between TRR and TAR is essential to the evaluation of αbecause a small α can result in the acceptance of too many erroneous classifications, whilst a large one may reject too many correct decisions. In this study, a novel evaluation metric Fit is proposed to consider both TRR and TAR:

$$\mathrm{Fit} = \mathrm{TAR} + \mathrm{TRR} - 1 \quad (9)$$

From Table II and Eq. (8) we can see that TRR and TAR correspond to specificity and sensitivity in binary classification, respectively. Thus Fit in Eq. (9) is equivalent to the Youden's J statistic, a commonly used measure of overall differentiation effectiveness in disease diagnoses [39], [40].

The error rate (Err) is related to user security in myoelectric control systems, hence it is also of great concern in rejection. For a CNN, Err can be simply equal to $1 - \mathrm{Acc}$, where Acc denotes the classification accuracy [23]:

$$\mathrm{Acc} = \frac{\text{number of correct classifications}}{\text{number of test samples}} \times 100\% \quad (10)$$

When $\mathrm{R}(\boldsymbol{p}^*, \alpha)$ is applied, part of the erroneous classifications will be rejected as no motion states, thus only those that are wrongly accepted by $\mathrm{R}(\boldsymbol{p}^*, \alpha)$ will be counted as errors. Referring to Table II, Err is revised as

$$\mathrm{Err} = \begin{cases} 1 - \mathrm{Acc} & \text{CNN} \\ \dfrac{\sum \mathrm{FA}}{\sum \mathrm{TA} + \sum \mathrm{FR} + \sum \mathrm{FA} + \sum \mathrm{TR}} & \text{CNN+Reject} \end{cases}. \quad (11)$$

F. Baseline Methods

To evaluate the effectiveness of the ConfScore model, two popular confidence features, i.e., the maximum posterior probability (MaxProb) and the entropy of the probability distribution [29], [30], are utilized as the baseline. In accordance with [30], MaxProb is normalized from $\left[\frac{1}{k}, 1\right]$ to [0, 1] for a fair comparison, where k is the number of classes. Considering that entropy is in general negatively related to classification accuracy, the inverse entropy (IEntropy) is defined as

$$\mathrm{IEntropy} = \frac{\log_2 m + \sum_{k=1}^{m} p_k \log_2 p_k}{\log_2 m} \quad (12)$$

III. Experiment Setup

A. Public Datasets

To evaluate the confidence estimation and rejection performance, six datasets of the NinaPro database (denoted as DB1-DB6, respectively) were utilized. DB1 was recorded using 10 Otto Bock 13-E200 electrodes . DB2, DB3 and DB6 were recorded using a Delsys Trigno wireless system. DB4 was recorded with a Cometa Wave Wireless sEMG system using Dormo SX-30 ECG electrodes. DB5 utilized the two Thalmic Myo armbands which is a low-cost device. More details can be found in Table III and [41], [42]. The sampling rates are 100 Hz for DB1, 200 Hz for DB5 and 2000 Hz for the other databases. These datasets have been widely applied in pilot studies for sEMG-based hand gesture classification [21]–[23], [36]. In terms of experiment protocols, DB1, DB2, DB4 and DB5 include more than 50 different hand or wrist movements of intact subjects. For example, 49 movements relevant to the activities of daily living are present in the DB2, including 8 isometric and isotonic hand configurations, 9 basic wrist movements, 23 grasping and functional movements and 9 force patterns. Different from these datasets, DB3 is composed of

TABLE II: Descriptions of TA/FA/FR/TR cases in rejection process.

		CNN Classifications	
		Correct	Erroneous
$\mathrm{R}\left(\boldsymbol{p}^{*}, \alpha\right)$	Accept	TA: accepted correct CNN estimations	FA: accepted erroneous CNN estimations
	Reject	FR: rejected correct CNN estimations	TR: rejected erroneous CNN estimations

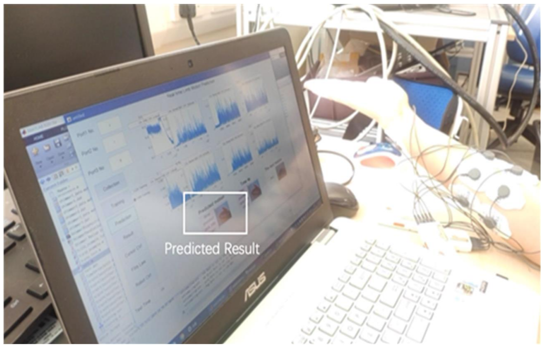

Fig. 2: Online testing using the customized platform

data collected from upper limb amputees, whilst the data of DB6 were recorded from 10 intact subjects repeating 7 grasps twice a day for 5 consecutive days [42].

B. Online Verification

To validate the effectiveness of ConfScore in real-time applications, a customized online platform was developed based on Shimmer wearable sensors and the Shimmer MATLAB Instrument Driver [43]. The platform was composed of several main modules: sEMG collection and streaming, data processing and plotting, CNN training, $\operatorname{ConfScore}\left(\boldsymbol{p}^{*}, \boldsymbol{\beta}\right)$ tuning, online classification and rejection analysis. The experiment involved six basic wrist/hand gestures: wrist flexion, wrist extension, supination, pronation, palm open, and palm close. Approved by the MaPS and Engineering joint Faculty Research Ethics Committee of University of Leeds, UK (reference MEEC 18-006), four healthy subjects (three male and one female, aged 20-55) took part in the experiment. As shown in Fig. 2, the participants were asked to perform predefined gestures following instructions given by the system. The ground-truth labels were created by requiring the user to hold each gesture for five seconds. Twelve bipolar electrodes were placed on the proximal portion of the left forearm to collect sEMG signals in six channels. The sampling rate of sEMG was set as 1024 Hz.

C. Data Pre-processing

DB1 provides a bandpass-filtered and Root-Mean-Square (RMS) rectified version of sEMG. DB4 was processed by a 10-Hz high-pass filter and a 1000 Hz low-pass filter. A Hampel filter was adopted to clean 50 Hz power-line interference from sEMG collected by the Delsys and Cometa sensors, i.e., DB2, DB3, DB4 and DB6. For DB5, Thalmic Myo incorporated a notch filter at 50 Hz. Based on filtered sEMG, a min–max normalization was implemented for each subject individually [44]. This normalization method was adopted since it can keep the original distribution of sEMG. To construct sEMG matrices for CNN, the window length was set as 300 ms with a 50 ms step for DB1 and DB5. By contrast, we empirically set 150 ms/50 ms for other databases. Herein the window length of DB1 and DB5 is comparatively larger because the sampling rates of these two databases are quite low (100 Hz and 200 Hz, respectively), thus the matrices constructed from shorter time windows could not support CNN. In the online platform, we adopted the 3rd order Butterworth band pass filter (20-450 Hz) and a 50 Hz notch filter for noise reduction. To construct the sEMG matrix for CNN, the window length and segmentation step were set as 150 ms and 25 ms, respectively.

D. Data Split

Following previous work, [21], [23], [36], [45], approximately two-thirds of the gesture trials in each subject of DB1-DB5 were utilized to train CNN and tune ConfScore subsequently. The remaining trials of the participant worked as the testing set to analyse confidence/rejection performances. Specifically, we set repetitions 2, 5 and 7 as the testing set in DB1; in DB2-DB5, we used repetitions 2 and 5 for testing. Since DB6 consists of data from multiple days, we trained CNN using data on the first day (Day 1), tuned ConfScore on Day 2, and tested the performances on Days 3-5.

E. Training of CNN

Hyper-parameters of CNN were first identified according to [23], [36] and then tuned empirically. Specifically, the network was trained in a 128-sized minibatch using stochastic gradient descent with momentum (SDGM). The momentum rate of SDGM was set as 0.9. The dynamic learning rate was initialized to be 0.0005. The dropout rate was 0.1 after every 10 epochs. The L2 regularization rate was set to be 0.01. We adopted 30 epochs for data training in DB1-DB5 and our online platform. The training data were shuffled in every epoch. To reduce overfitting, the dropout rate was set as 0.2 for DB1-DB5. We adopted fewer epochs (20) and a higher dropout rate (0.5) for DB6 due to the domain variation among training and testing sessions in different days.

F. Training of ConfScore

In $\operatorname{ConfScore}\left(\boldsymbol{p}^{*}, \boldsymbol{\beta}\right)$ we can define left and right edges of our activation function flexibly by adjusting γ_1 and γ_2. Empirically, we find that an explicit tuning of γ_1 and γ_2 for each dataset can further optimize confidence estimation. For the sake of simplicity and generalization, in this study we kept γ_1 and γ_2 fixed as $\{\gamma_1 = 0, \gamma_2 = 1\}$ for all trials in six databases and the online platform. In addition, the GA was utilized to search $\hat{\boldsymbol{\beta}}$ by exploiting reproduction, crossover, and mutation operators. In our implementation, an elitist strategy was further incorporated in this algorithm to enhance convergence.

TABLE III: Specifications of the public databases used in this paper.

Database	No. of gestures	Major gestures	Upper limb amputation	No. of channels	Devices
DB1	52	finger/wrist/grasping movements	healthy	10	10 Otto Bock 13-E200, 100Hz
DB2	49	finger/wrist/grasping movements force patterns	healthy	12	Delsys Trigno wireless system, 2000Hz
DB3	49	finger/wrist/grasping movements force patterns	amputated	12	Delsys Trigno wireless system, 2000Hz
DB4	52	finger/wrist/grasping movements	healthy	12	Cometa Wave Wireless sEMG system, 2000Hz
DB5	52	finger/wrist/grasping movements	healthy	16	Thalmic Myo armbands, 200Hz
DB6	7	grasping movements	healthy	16	Delsys Trigno wireless system, 2000Hz

G. Statistical Analysis

In this study, statistical analysis was performed using the Statistics and Machine Learning Toolbox in MATLAB R2012a. In particular, the assumption of data normality was first checked via the Shapiro–Wilk test (the level was set to be 0.05) [46]. If the assumption was satisfied, the one-way analysis of variance (ANOVA) test was applied to verify the differences in methods of confidence estimation and rejection process; otherwise, its nonparametric equivalent, i.e., Kruskal–Wallis (KW) test, was performed alternatively.

IV. Results

A. Distribution of Confidence Features

As suggested in [30], distributions of confidence features of a classifier are expected to be indicative of classification accuracy, i.e., correct classifications are with high scores (close to one) whereas wrong predictions result in lower scores (close to zero). Fig. 3 visualizes the distributions of correct and erroneous classifications following three different confidence features. As we can see, correct classifications are overwhelmingly gathering in the range of [0.95, 1] when ConfScore is utilized. In opposite, erroneous classifications are gathering mainly in bins of smaller ConfScore. Differently, distributions between correct and erroneous classifications tend to be less distinguishable when MaxProb or IEntropy is utilized. Therefore, we infer that ConfScore can be more relevant with CNN confidence in terms of the classification accuracy. In the following sections we will explore how the distribution differences further influence BMEC and Fit of three confidence features.

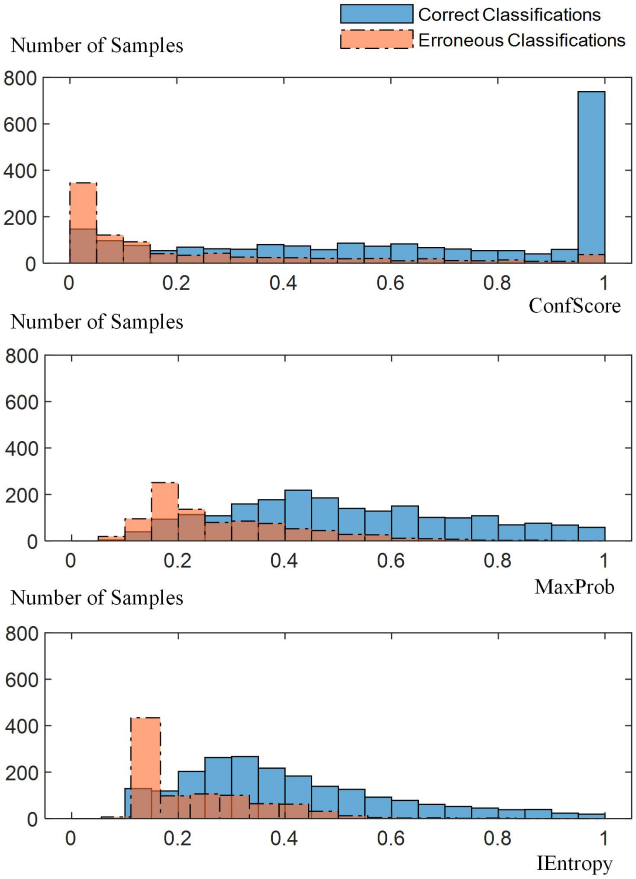

Fig. 3: Distributions of correct and erroneous classifications in testing sets of Subject 2-DB1. The width of a bin is 0.05 and the amplitude of each bin denotes the number of samples (CNN outputs) whose confidence features are located in the corresponding range.

B. BMEC of Confidence Features

Table IV lists BMEC of ConfScore, MaxProb and IEntropy in six public databases. In summary, BMEC of ConfScore is larger than MaxProb and IEntropy for most subjects in each database. From Table IV we can also see that the average of BMEC of all three confidence features in DB1, DB2, DB4 and DB5 are comparatively larger than the average of all three confidence features in DB3, DB6 because confidence distributions of correct and erroneous classifications are less distinguishable in DB3 and DB6. The deterioration occurs since the experimental protocols in DB3 and DB6 are more challenging. In terms of DB3, it is hard for trans-radial amputees to produce reliable ground truth because of the inability to operate any sensor on the missing limbs [45]. In DB6, both CNN and ConfScore are trained and tested in different days, where the electrode shift can have severe impacts on model accuracies.

C. Analysis of Rejection Process

In this section we analyse the effect of $\mathrm{R}\left(\boldsymbol{p}^{*}, \alpha\right)$ on the basis of ConfScore. Referring to Eq. (7) and Table II, given a specific α, Err, TAR, TRR and Fit can be calculated accordingly. Fig. 4 shows variations of these four metrics following a changing α in the testing set of Subject 1-DB4. The variation step for α is set to be 0.05. Fig. 4 shows that

TABLE IV: BMEC of ConfScore, MaxProb and IEntropy for all subjects in six databases. SD denotes the standard deviation.

Database	Confidence Feature	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	Mean	SD
DB1	ConfScore	**0.39**	**0.41**	**0.36**	**0.34**	**0.29**	**0.34**	**0.3**	**0.38**	**0.28**	**0.35**	**0.34**	0.04
	MaxProb	0.31	0.34	0.3	0.3	0.23	0.27	0.26	0.3	0.22	0.29	0.28	0.04
	IEntropy	0.27	0.3	0.25	0.25	0.2	0.23	0.22	0.27	0.19	0.25	0.24	0.03
DB2	ConfScore	**0.47**	**0.43**	**0.31**	**0.29**	**0.29**	**0.33**	**0.22**	**0.5**	**0.38**	**0.37**	**0.36**	0.09
	MaxProb	0.34	0.3	0.21	0.22	0.21	0.24	0.17	0.36	0.25	0.23	0.25	0.06
	IEntropy	0.27	0.24	0.16	0.18	0.15	0.18	0.13	0.3	0.2	0.18	0.2	0.05
DB3	ConfScore	**0.22**	**0.31**	**0.69**	**0.14**	**0.08**	**0.12**	0.03	**0.31**	**0.33**	**0.09**	**0.23**	0.19
	MaxProb	0.16	0.22	0.56	0.12	0.06	0.09	**0.04**	0.24	0.23	0.07	0.18	0.15
	IEntropy	0.13	0.15	0.55	0.1	0.05	0.06	0.01	0.19	0.17	0.06	0.15	0.15
DB4	ConfScore	**0.57**	**0.42**	**0.22**	**0.66**	**0.63**	**0.42**	**0.69**	**0.65**	**0.48**	**0.47**	**0.52**	0.15
	MaxProb	0.43	0.3	0.18	0.46	0.46	0.32	0.5	0.48	0.37	0.36	0.39	0.10
	IEntropy	0.4	0.23	0.13	0.42	0.42	0.26	0.47	0.45	0.31	0.32	0.34	0.11
DB5	ConfScore	**0.62**	**0.6**	**0.64**	**0.64**	**0.62**	**0.58**	**0.61**	**0.57**	**0.64**	**0.62**	**0.61**	0.04
	MaxProb	0.52	0.47	0.5	0.53	0.47	0.46	0.48	0.42	0.45	0.56	0.49	0.04
	IEntropy	0.47	0.4	0.48	0.48	0.45	0.43	0.45	0.37	0.41	0.51	0.44	0.04
DB6 (Day3)	ConfScore	**0.28**	**0.03**	**0.22**	**-0.02**	**0.18**	**0.17**	**0.62**	**0.1**	**0.11**	**0.39**	**0.21**	0.19
	MaxProb	0.14	0.01	0.1	-0.05	0.06	0.16	0.41	0.07	0.1	0.22	0.12	0.13
	IEntropy	0.11	0.03	0.07	-0.07	0.05	0.16	0.41	0.09	0.07	0.19	0.11	0.13
DB6 (Day4)	ConfScore	**0.39**	**0.3**	**0.01**	**0.04**	**0.13**	**0.22**	**0.34**	**0.27**	**0.26**	**0.11**	**0.21**	0.13
	MaxProb	0.22	0.21	-0.01	0.02	0.06	0.14	0.19	0.19	0.17	0.06	0.13	0.08
	IEntropy	0.21	0.19	-0.01	0	0.06	0.11	0.18	0.15	0.17	0.07	0.11	0.08
DB6 (Day5)	ConfScore	**0.32**	**0.24**	**0.18**	**0.23**	**0.4**	**0.33**	**0.22**	**0.19**	**0.18**	**0.38**	**0.27**	0.08
	MaxProb	0.17	0.15	0.16	0.12	0.3	0.28	0.13	0.19	0.12	0.2	0.18	0.06
	IEntropy	0.12	0.13	0.17	0.08	0.26	0.3	0.14	0.15	0.12	0.17	0.16	0.07

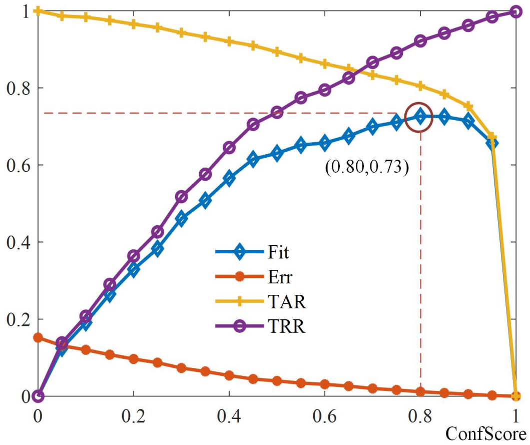

Fig. 4: Err, TAR, TRR and Fit for various rejection thresholds in testing sets of Subject 1-DB4.

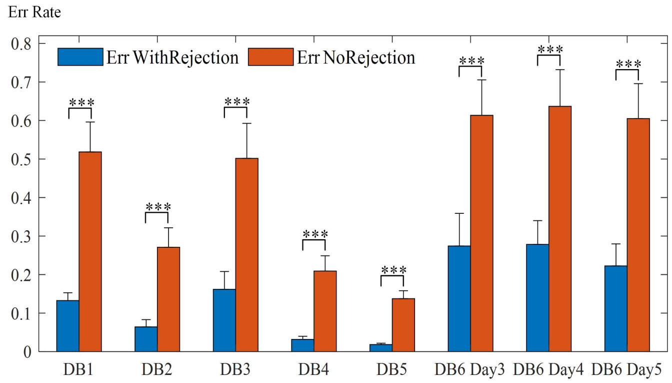

Fig. 5: Statistical analysis of Err for testing sets with and without rejection in all databases (***p-value < 0.001, **p-value < 0.01, *p-value < 0.05).

Err is decreasing monotonically along with α. When $\alpha = 0$, there is no rejection cascaded with CNN, thus $\mathrm{Err} = 1 - \mathrm{Acc}$. By contrast, Err becomes zero when $\alpha = 1$, since all CNN decisions are rejected in this case.

Since TAR decreases monotonically and TRR increases inversely, a concave downward curve of Fit is thus obtained. More specifically, Fit increases continuously when α is comparatively smaller, and there comes a turning point (the aubergine circles in Fig. 4) when α becomes larger. Hence, focusing on Err or TRR alone can result in an decreased TAR, verifying the necessity of Fit for the trade-off between TRR and TAR in rejection analysis. When the rejection is conducted around the turning point shown in Fig. 4, we can obtain a smaller Err (1.2% for Subject 1-DB4) with an acceptable TAR (80.5%).

To further illustrate the effectiveness of Fit in rejection, we apply the optimal α of training sets to the corresponding testing sets in each subject. Fig. 5 compares Err with and without rejection for testing sets in all databases. Moreover, Fig. 6 shows statistical results of TAR and TRR in Fit of all databases. In this study, we conducted ANOVA/KW for statistical analysis. The feature factor has two levels (Rejection/NoRejection for Fig. 5, TAR/TRR for Fig. 6), and subjects of each database work as the random variable. From Fig. 5 we can see that in each database Err is reduced significantly by rejecting low confidence classifications. Moreover, since Fit attempts to compromise TAR and TRR, these two metrics can be close to each other in most databases. Taking DB5 as example, the mean value of TAR is 0.83 whilst TRR reaches 0.9 on average, indicating that the majority of erroneous classifications can be rejected whilst causing small loss on correct classifications, indicating a promising trade-off between system security and rejection cost.

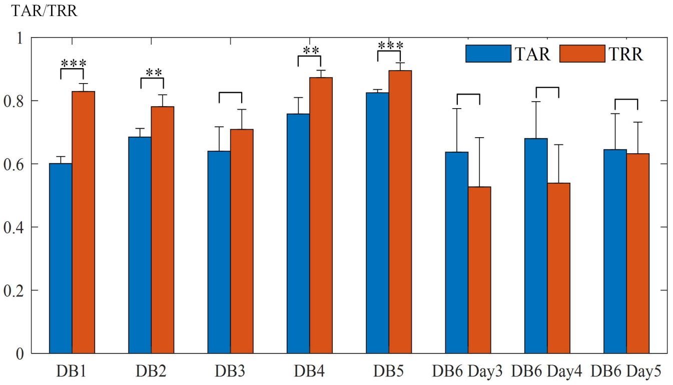

Fig. 6: Statistical analysis of TAR and TRR for testing sets in all databases (***p-value < 0.001, **p-value < 0.01, *p-value < 0.05).

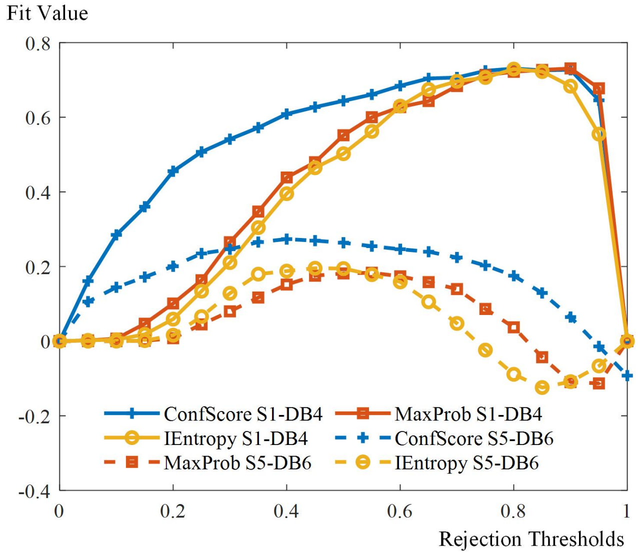

Fig. 7: Fit values for various rejection thresholds when three confidence features are used in testing sets of Subject1-DB4 and Subject 5-DB6.

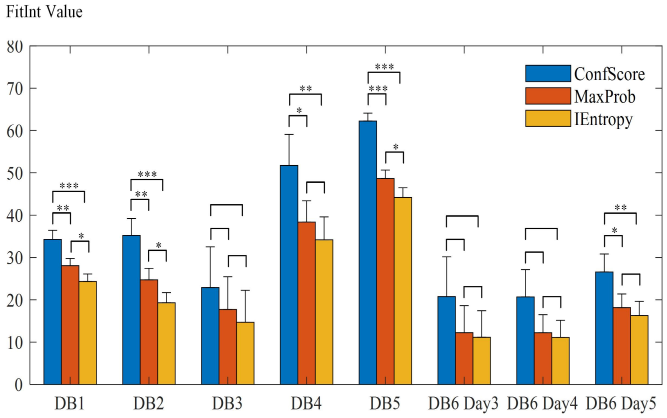

Fig. 8: Statistical analysis of FitInt values of rejection in all databases when three confidence features are adopted (***p-value < 0.001, **p-value < 0.01, *p-value < 0.05).

TABLE V: BMEC of ConfScore, MaxProb and IEntropy in online testing.

Confidence Feature	S1	S2	S3	S4	Mean	SD
ConfScore	0.45	0.53	0.38	0.28	0.41	0.11
MaxProb	0.22	0.26	0.12	0.13	0.18	0.06
IEntropy	0.18	0.20	0.05	0.07	0.13	0.08

D. *Comparison of Confidence Features in Rejection*

Fig. 7 shows Fit values for various rejection thresholds in testing sets of Subject1-DB4 and Subject 5-DB6. Recall that Fit can be smaller than zero, in which case $\mathrm{R}(\boldsymbol{p}^*, \alpha)$ either accepts too many erroneous CNN decisions or rejects too many correct ones. From this figure, several interesting results can be observed. First, for each confidence feature the Fit of α in Subject1-DB4 are in general much higher than in Subject 5-DB6. This observation will be further discussed in Section V.B. Second, although the maximal Fit of three confidence features can be close in some cases (such as Subject1-DB4), the Fit curves of ConfScore are always much wider and more flattened than the curves of two other features. This characteristic means that when adopting ConfScore as a CNN confidence feature, Fit is less sensitive to variations of α in the rejection process, which contributes to a wider range for threshold selection. In the following part, this characteristic is denoted as the rejection robustness for Fit.

To further quantify the rejection robustness, we calculate the integral of Fit curves (denoted as FitInt) using the trapezoidal method. Fig. 8 illustrates the mean and standard deviations of FitInt for all subjects in each database. As we can see, FitInt of ConfScore is much larger than FitInt of MaxProb and IEntropy in most databases. Thus, the rejection performance can be less sensitive to threshold variations when ConfScore is used as the confidence feature. In addition, we can also observe that FitInt values of three confidence features in DB3 and DB6 are with smaller means but larger standard deviations. These degradations are consistent with BMEC performance shown in Table IV.

E. *Online Results*

Table V and Table VI list the BMEC and FitInt of three confidence features in online testing. These tables show that ConfScore outperforms MaxProb and IEntropy significantly in both confidence distribution and rejection robustness (as for BMEC, the p-value of ConfScore versus MaxProb/IEntropy is 0.011/0.005; as for FitInt, the p-value is 0.003/0.002). These outcomes are consistent with results of offline analysis (Table IV and Fig. 8). Moreover, since the confidence-based rejection can be regarded as a post-processing method, we also compare the performance of our approach with a widely applied smoothing strategy, i.e. Majority Vote (MV) [47]. Table VII lists the Err of CNN, CNN+MV and CNN+Rejection for four subjects in online testing. Specifically, the ConfScore works as the confidence feature in rejection, whilst MV makes the final decision using the current classification result along with six previous results. As shown in Table VII, although MV can further reduce Err in classification, its performance is evidently worse than the performance of the confidence-based rejection because the final result of MV is still an active motion rather than a no-motion state. Compared with MV, the rejection method is more useful when the control security is crucial, particularly in human-machine interface.

TABLE VI: FitInt of ConfScore, MaxProb and IEntropy in online testing.

Confidence Feature	S1	S2	S3	S4	Mean	SD
ConfScore	40.35	49.76	39.92	29.66	39.92	8.21
MaxProb	18.95	23.93	13.82	13.65	17.59	4.89
IEntropy	14.81	18.52	5.37	6.62	11.33	6.36

TABLE VII: Err of CNN, CNN+MV and CNN+Rejection in online testing. The ConfScore works as the confidence feature in rejection.

Method	S1	S2	S3	S4	Mean	SD
CNN	0.26	0.28	0.32	0.37	0.31	0.05
CNN+MV	0.24	0.25	0.31	0.34	0.29	0.05
CNN+Rejection	0.07	0.07	0.10	0.15	0.10	0.04

TABLE VIII: Average BMEC of ConfScore in each database using different pairs of γ_1 and γ_2.

Parameter Pairs	DB1	DB2	DB3	DB4	DB5	DB6
γ_1=0, γ_2=0.2	0.45	0.45	0.32	0.58	0.72	0.22
γ_1=0, γ_2=0.4	0.44	0.44	0.33	0.58	0.71	0.23
γ_1=0, γ_2=0.6	0.41	0.41	0.28	0.56	0.69	0.26
γ_1=0, γ_2=0.8	0.34	0.35	0.23	0.53	0.67	0.24

V. Discussion

In real-time applications, a deep network normally performs well in its trained datasets but tends to fail in unseen ones [48]. In this study, a preliminary research was conducted to estimate model confidence for CNN-based hand gesture recognition, which helps to improve the reliability of myoelectric control. To be specific, we propose a novel confidence estimation model $\mathrm{ConfScore}\,(\boldsymbol{p}^*, \boldsymbol{\beta})$ to generate confidence scores based on posterior probabilities of CNN. Following a threshold-based rejection rule, unconfident classifications can be identified and rejected. In addition, although the main target of rejection is to refuse erroneous classifications and improve system security, focusing on this target alone may result in a serious control lag by setting overlarge rejection thresholds arbitrarily. To address this issue, we propose Fit which combines TAR and TRR, so that both system security and control lag can be fully considered in threshold selection or rejection analysis.

A. *Design of* $\mathrm{ConfScore}\,(\boldsymbol{p}^*, \boldsymbol{\beta})$

In the design of $\mathrm{ConfScore}\,(\boldsymbol{p}^*, \boldsymbol{\beta})$, we utilized the zero-order smooth-step function to map posterior likelihood of CNN to associated confidence features. A main advantage is that we can define left and right edges of the activation function flexibly by adjusting γ_1 and γ_2. Table VIII lists the average BMEC of subjects in six databases when four different pairs of γ_1 and γ_2 are utilized. As we can see, an explicit tuning of γ_1 and γ_2 can further optimize confidence distributions in each database. Another novel design is the BMEC-based objective function to train $\mathrm{ConfScore}\,(\boldsymbol{p}^*, \boldsymbol{\beta})$ in a supervised way. As discussed in [30], the confidence feature is expected to be closer to one when CNN is certain about the decision (i.e., the classification is prone to be correct) and to be near zero when CNNs are making uncertain decisions. As summarized in Table IV, for most cases of six datasets, BMEC values of ConfScore are comparatively larger than the values of MaxProb and IEntropy. Therefore, we infer that the proposed ConfScore can better reflect correctness of CNN classifications.

B. *Design of Fit and FitInt*

As illustrated in Fig. 1, given the ConfScore of classification results, a threshold-based rejection can be utilized to improve the control reliability. To illustrate this process, Fig. 9 compares the confusion matrix without/with rejection. As we can see, in the conventional classification without rejection, a high error rate is obtained in the case presented. Differently, by identifying classifications whose ConfScore values are smaller than a predetermined threshold, a large number of erroneous classifications can be identified and rejected. This is how ConfScore can help to improve the robustness of gesture classification. However, from Fig. 9 we can also observe that some of the correct classifications (those in the diagonal of the confusion matrix without rejection) are rejected mistakenly. These mistakes should be regarded as the cost of the rejection process. Therefore, as illustrated in Eq. (9) and Table II, we have proposed Fit to conduct a quantitative evaluation of rejection performance. From Fig. 4 we can see that Fit compromises TAR and TRR to achieve a good balance between control continuity and system security.

In this study we also compare the rejection robustness of different confidence features (see Fig. 8). Our concern are: 1) as shown in Fig. 7, the selection of the rejection threshold can affect the Fit value; 2) the threshold determination is usually made empirically based on previous datasets, and this strategy can be affected by the inconsistency of Fit curves between datasets. Fig. 10 lists the optimal thresholds for three confidence features using data of DB6 (Day4) and DB6 (Day5). Apparently, for most subjects the optimal thresholds in Day4 and Day5 are very different; and the thresholds determined based on previous datasets may result in a poor Fit for the target dataset. Based on these observations, we infer that it is important to compare robustness, i.e. FitInt, which can help to indicate how the Fit value is robust to variations of threshold. As shown in Fig. 8, FitInt of ConfScore is much larger than the FitInt of MaxProb and IEntropy in most databases. Similarly, previous research [16], [19] also suggested that a desirable confidence characteristic should leave a wider range for threshold adjustment.

As mentioned in Section IV.D, an interesting observation in Fig. 7 is the difference of Fit curves among datasets. Specifically, for each confidence feature, Fit curve of Subject1-DB4 is always higher than the Fit curve of Subject 5-DB6, which indicates a better rejection performance of the former participant. According to Table IV, BMEC values of three confidence features in Subject 5-DB6 are smaller than the BMEC values of three confidence features in Subject1-DB4. Based on Eq. (5), we know that the distributions of correct/erroneous classifications in Subject 5-DB6 are less distinguishable. A main reason is that the CNN classifier is trained and tested using data of different days in DB6, thus the confidence features could become less qualified due to the

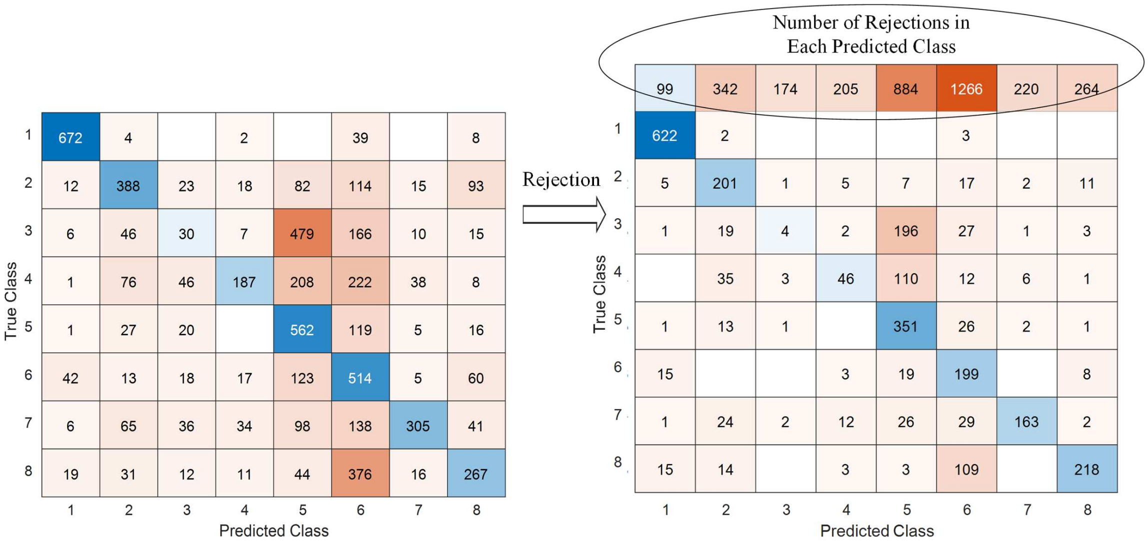

Fig. 9: Confusion matrix without/with rejection based on ConfScore. The presented cased is Subject 1 of DB6. The optimal rejection threshold is pre-determined based on the training data. The error rate of conventional classification is high since training and testing datasets are from different days of a subject. The Fit of the rejection performance is 0.36.

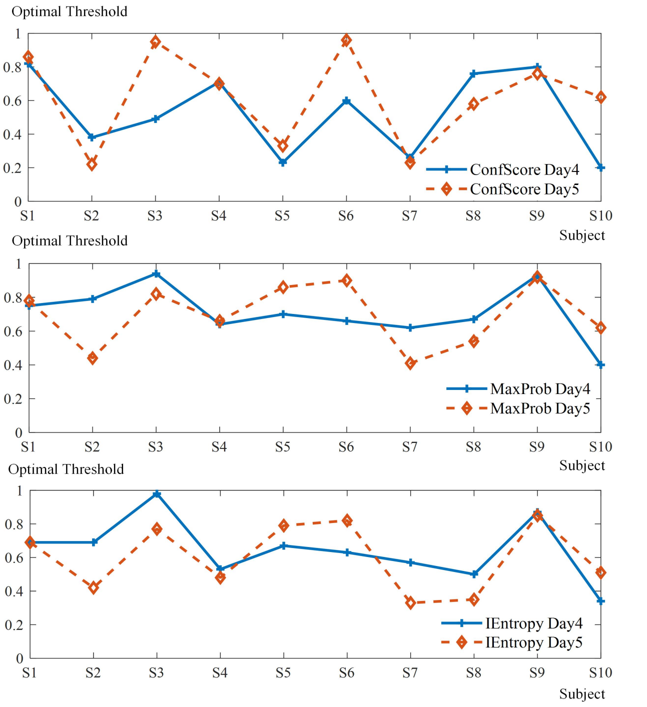

Fig. 10: Optimal thresholds for Fit curves of three confidence features in Day4 and Day5 of DB6.

degradation of CNN accuracy.

VI. Conclusions and Future Work

In this study, we introduced a preliminary attempt to estimate CNN confidence for rejection-based hand gesture classification in myoelectric control systems. By analysing posterior likelihood of softmax layer, the proposed confidence model can provide scores (ConfScore) highly related to correctness of CNN predictions. The superiority of ConfScore to two commonly utilized confidence features is fully verified via analysis of BMEC and rejection robustness using data from public databases and our online experiments. With help of confidence-based rejection, the error rate of CNN can be reduced significantly with small loss of correct classifications, thereby enhancing the model reliability in sEMG-based gesture recognition.

Since this study was preliminary research on CNN confidence estimation and its application in rejection, there remain some interesting and open questions which deserve further investigation. For instance, TARs in DB6 are observed to be bigger than in TRR, but this trend is inverted in other databases. We guess that the reason for the specialty of DB6 could be domain shift, since training and testing data are from different days of a subject. Thus, more variety of datasets should be involved for further verification. In addition, only conventional gesture recognition is involved. In the future work, we anticipate conducting further investigations involving various tasks, such as the Box and Blocks Task, Cups Relocation Task, and Block Turn Task, etc. Furthermore, we will also investigate the model confidence of regression approaches which are fundamental for simultaneous and proportional estimation of joint kinematics.

References

[1] W. Yang, D. Yang, Y. Liu, and H. Liu, "Decoding simultaneous multi-dof wrist movements from raw emg signals using a convolutional neural network," *IEEE Trans. Human-Mach. Syst.*, vol. 49, no. 5, pp. 411–420, 2019.

[2] A. Furui, S. Eto, K. Nakagaki, K. Shimada, G. Nakamura, A. Masuda, T. Chin, and T. Tsuji, "A myoelectric prosthetic hand with muscle synergy–based motion determination and impedance model–based biomimetic control," *Science Robotics*, vol. 4, no. 31, p. eaaw6339, 2019.

[3] J. M. Hahne, M. A. Schweisfurth, M. Koppe, and D. Farina, "Simultaneous control of multiple functions of bionic hand prostheses: Performance and robustness in end users," *Sci. Robot.*, vol. 3, no. 19, p. eaat3630, 2018.

[4] W. Guo, X. Sheng, H. Liu, and X. Zhu, "Toward an enhanced human–machine interface for upper-limb prosthesis control with combined emg and nirs signals," *IEEE Trans. Human-Mach. Syst.*, vol. 47, no. 4, pp. 564–575, 2017.

[5] J. Maier, A. Naber, and M. Ortiz-Catalan, "Improved prosthetic control based on myoelectric pattern recognition via wavelet-based de-noising," *IEEE Trans. Neural Syst. Rehabilitation Eng.*, vol. 26, no. 2, pp. 506–514, 2017.

[6] R. N. Khushaba, A. H. Al-Timemy, A. Al-Ani, and A. Al-Jumaily, "A framework of temporal-spatial descriptors-based feature extraction for improved myoelectric pattern recognition," *IEEE Trans. Neural Syst. Rehabilitation Eng.*, vol. 25, no. 10, pp. 1821–1831, 2017.

[7] J. Qi, G. Jiang, G. Li, Y. Sun, and B. Tao, "Surface emg hand gesture recognition system based on pca and grnn," *NEURAL COMPUT APPL*, pp. 1–9, 2019.

[8] J. He, X. Sheng, X. Zhu, C. Jiang, and N. Jiang, "Spatial information enhances myoelectric control performance with only two channels," *IEEE Trans. Ind. Informat.*, vol. 15, no. 2, pp. 1226–1233, 2018.

[9] M. Rossi, S. Benatti, E. Farella, and L. Benini, "Hybrid emg classifier based on hmm and svm for hand gesture recognition in prosthetics," in *2015 IEEE International Conference on Industrial Technology (ICIT)*. IEEE, 2015, pp. 1700–1705.

[10] B. Yu, X. Zhang, L. Wu, X. Chen, and X. Chen, "A novel postprocessing method for robust myoelectric pattern-recognition control through movement pattern transition detection," *IEEE Trans. Human-Mach. Syst.*, vol. 50, no. 1, pp. 32–41, 2019.

[11] X. Sheng, B. Lv, W. Guo, and X. Zhu, "Common spatial-spectral analysis of emg signals for multiday and multiuser myoelectric interface," *Biomed Signal Process Control*, vol. 53, p. 101572, 2019.

[12] A. Waris, I. K. Niazi, M. Jamil, K. Englehart, W. Jensen, and E. N. Kamavuako, "Multiday evaluation of techniques for emg based classification of hand motions," *IEEE journal of biomedical and health informatics*, 2018.

[13] E. C. Hill, T. J. Housh, C. Smith, K. C. Cochrane, N. Jenkins, J. T. Cramer, R. J. Schmidt, and G. Johnson, "Effect of sex on torque, recovery, emg, and mmg responses to fatigue," *J Musculoskelet Neuronal Interact*, vol. 16, no. 4, p. 310, 2016.

[14] L. Pan, D. Zhang, N. Jiang, X. Sheng, and X. Zhu, "Improving robustness against electrode shift of high density emg for myoelectric control through common spatial patterns," *J NEUROENG REHABIL*, vol. 12, no. 1, p. 110, 2015.

[15] R. Kusche and M. Ryschka, "Combining bioimpedance and emg measurements for reliable muscle contraction detection," *IEEE Sens. J.*, vol. 19, no. 23, pp. 11 687–11 696, 2019.

[16] J. Robertson, E. Scheme, and K. Englehart, "Effects of confidence-based rejection on usability and error in pattern recognition-based myoelectric control," *IEEE journal of biomedical and health informatics*, 2018.

[17] E. J. Scheme, B. S. Hudgins, and K. B. Englehart, "Confidence-based rejection for improved pattern recognition myoelectric control," *IEEE Trans. Biomed. Eng.*, vol. 60, no. 6, pp. 1563–1570, 2013.

[18] S. Amsüss, P. M. Goebel, N. Jiang, B. Graimann, L. Paredes, and D. Farina, "Self-correcting pattern recognition system of surface emg signals for upper limb prosthesis control," *IEEE Trans. Biomed. Eng.*, vol. 61, no. 4, pp. 1167–1176, 2013.

[19] E. Scheme and K. Englehart, "A comparison of classification based confidence metrics for use in the design of myoelectric control systems," in *2015 37th annual international conference of the IEEE engineering in medicine and biology society (EMBC)*. IEEE, 2015, pp. 7278–7283.

[20] A. Ameri, M. A. Akhaee, E. Scheme, and K. Englehart, "Regression convolutional neural network for improved simultaneous emg control," *J. Neural Eng.*, vol. 16, no. 3, p. 036015, 2019.

[21] W. Wei, Y. Wong, Y. Du, Y. Hu, M. Kankanhalli, and W. Geng, "A multi-stream convolutional neural network for semg-based gesture recognition in muscle-computer interface," *Pattern Recognit. Lett.*, 2017.

[22] Z. Ding, C. Yang, Z. Tian, C. Yi, Y. Fu, and F. Jiang, "semg-based gesture recognition with convolution neural networks," *Sustainability*, vol. 10, no. 6, p. 1865, 2018.

[23] W. Wei, Q. Dai, Y. Wong, Y. Hu, M. Kankanhalli, and W. Geng, "Surface electromyography-based gesture recognition by multi-view deep learning," *IEEE Trans. Biomed. Eng.*, 2019.

[24] Y. Yamanoi, Y. Ogiri, and R. Kato, "Emg-based posture classification using a convolutional neural network for a myoelectric hand," *Biomed Signal Process Control*, vol. 55, p. 101574, 2020.

[25] X. Zhai, B. Jelfs, R. H. Chan, and C. Tin, "Self-recalibrating surface emg pattern recognition for neuroprosthesis control based on convolutional neural network," *Front. Neurosci.*, vol. 11, p. 379, 2017.

[26] V. Shanmuganathan, H. R. Yesudhas, M. S. Khan, M. Khari, and A. H. Gandomi, "R-cnn and wavelet feature extraction for hand gesture recognition with emg signals," *NEURAL COMPUT APPL*, vol. 32, no. 21, pp. 16 723–16 736, 2020.

[27] R. Ranjan, V. M. Patel, and R. Chellappa, "Hyperface: A deep multi-task learning framework for face detection, landmark localization, pose estimation, and gender recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 1, pp. 121–135, 2017.

[28] C. Zhang, X. Pan, H. Li, A. Gardiner, I. Sargent, J. Hare, and P. M. Atkinson, "A hybrid mlp-cnn classifier for very fine resolution remotely sensed image classification," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 140, pp. 133–144, 2018.

[29] X. Wang, Y. Luo, D. Crankshaw, A. Tumanov, F. Yu, and J. E. Gonzalez, "Idk cascades: Fast deep learning by learning not to overthink," *arXiv preprint arXiv:1706.00885*, 2017.

[30] S. Wan, T.-Y. Wu, W. H. Wong, and C.-Y. Lee, "Confnet: Predict with confidence," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 2921–2925.

[31] H. Li, Z. Lin, X. Shen, J. Brandt, and G. Hua, "A convolutional neural network cascade for face detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 5325–5334.

[32] Y. Hu, Y. Wong, W. Wei, Y. Du, M. Kankanhalli, and W. Geng, "A novel attention-based hybrid cnn-rnn architecture for semg-based gesture recognition," *PloS one*, vol. 13, no. 10, p. e0206049, 2018.

[33] Y. LeCun *et al.*, "Lenet-5, convolutional neural networks," *URL: http://yann. lecun. com/exdb/lenet*, vol. 20, p. 5, 2015.

[34] W. Geng, Y. Du, W. Jin, W. Wei, Y. Hu, and J. Li, "Gesture recognition by instantaneous surface emg images," *Sci. Rep.*, vol. 6, p. 36571, 2016.

[35] Y. Du, W. Jin, W. Wei, Y. Hu, and W. Geng, "Surface emg-based inter-session gesture recognition enhanced by deep domain adaptation," *Sensors*, vol. 17, no. 3, p. 458, 2017.

[36] M. Atzori, M. Cognolato, and H. Müller, "Deep learning with convolutional neural networks applied to electromyography data: A resource for the classification of movements for prosthetic hands," *Frontiers in neurorobotics*, vol. 10, p. 9, 2016.

[37] Z. Tayeb, N. Waniek, J. Fedjaev, N. Ghaboosi, L. Rychly, C. Widderich, C. Richter, J. Braun, M. Saveriano, G. Cheng *et al.*, "Gumpy: A python toolbox suitable for hybrid brain–computer interfaces," *J. Neural Eng.*, vol. 15, no. 6, p. 065003, 2018.

[38] Y.-J. Gong, J.-J. Li, Y. Zhou, Y. Li, H. S.-H. Chung, Y.-H. Shi, and J. Zhang, "Genetic learning particle swarm optimization," *IEEE transactions on cybernetics*, vol. 46, no. 10, pp. 2277–2290, 2015.

[39] E. F. Schisterman, N. J. Perkins, A. Liu, and H. Bondell, "Optimal cut-point and its corresponding youden index to discriminate individuals using pooled blood samples," *Epidemiology*, pp. 73–81, 2005.

[40] T. Mazzu-Nascimento, G. G. Morbioli, L. A. Milan, F. C. Donofrio, C. A. Mestriner, and E. Carrilho, "Development and statistical assessment of a paper-based immunoassay for detection of tumor markers," *Analytica chimica acta*, vol. 950, pp. 156–161, 2017.

[41] S. Pizzolato, L. Tagliapietra, M. Cognolato, M. Reggiani, H. Müller, and M. Atzori, "Comparison of six electromyography acquisition setups on hand movement classification tasks," *PloS one*, vol. 12, no. 10, p. e0186132, 2017.

[42] F. Palermo, M. Cognolato, A. Gijsberts, H. Müller, B. Caputo, and M. Atzori, "Repeatability of grasp recognition for robotic hand prosthesis control based on semg data," in *2017 International Conference on Rehabilitation Robotics (ICORR)*. IEEE, 2017, pp. 1154–1159.

[43] A. Burns, B. R. Greene, M. J. McGrath, T. J. O'Shea, B. Kuris, S. M. Ayer, F. Stroiescu, and V. Cionca, "Shimmer™–a wireless sensor platform for noninvasive biomedical research," *IEEE Sens. J.*, vol. 10, no. 9, pp. 1527–1534, 2010.

[44] Y. Wan, Z. Han, J. Zhong, and G. Chen, "Pattern recognition and bionic manipulator driving by surface electromyography signals using convolutional neural network," *International Journal of Advanced Robotic Systems*, vol. 15, no. 5, p. 1729881418802138, 2018.

[45] M. Atzori, A. Gijsberts, C. Castellini, B. Caputo, A.-G. M. Hager, S. Elsig, G. Giatsidis, F. Bassetto, and H. Müller, "Electromyography data for non-invasive naturally-controlled robotic hand prostheses," *Scientific data*, vol. 1, p. 140053, 2014.

[46] A. Gallina, R. Merletti, and M. Gazzoni, "Uneven spatial distribution of surface emg: what does it mean?" *Eur. J. Appl. Physiol.*, vol. 113, no. 4, pp. 887–894, 2013.

[47] A. D. Chan and G. C. Green, "Myoelectric control development toolbox," *CMBES Proceedings*, vol. 30, 2007.

[48] C. Duan, S. Junginger, J. Huang, K. Jin, and K. Thurow, "Deep learning for visual slam in transportation robotics: a review," *Transp. Saf. Environ.*, vol. 1, no. 3, pp. 177–184, 2019.