A Transfer Learning Model for Gesture Recognition Based on the Deep Features Extracted by CNN

Yongxiang Zou and Long Cheng, *Senior Member, IEEE*

***Abstract*—The surface electromyogram (sEMG) based hand gesture recognition is prevalent in human–computer interface systems. However, the generalization of the recognition model does not perform well on cross-subject and cross-day. Transfer learning, which applies the pretrained model to another task, has demonstrated its effectiveness in solving this kind of problem. In this regard, this article first proposes a multiscale kernel convolutional neural network (MKCNN) model to extract and fuse multiscale features of the multichannel sEMG signals. Based on the proposed MKCNN model, a transfer learning model named TL-MKCNN combines the MKCNN and its Siamese network by a custom distribution normalization module (DNM) and a distribution alignment module (DAM) to realize domain adaptation. The DNM can cluster the deep features extracted from different domains to their category center points embedded in the feature space, and the DAM further aligns the overall distribution of the deep features from different domains. The MKCNN model and the TL-MKCNN model are tested on various benchmark databases to verify the effectiveness of the transfer learning framework. The experimental results show that, on the benchmark database NinaPro DB6, the average accuracies of TL-MKCNN can achieve 97.22% on within-session, 74.48% on cross-subject, and 90.30% on cross-day, which are 4.31%, 11.58%, and 5.51% higher than those of the MKCNN model on within-session, cross-subject, and cross-day, respectively. Compared with the state-of-the-art works, the TL-MKCNN obtains 13.38% and 37.88% accuracy improvement on cross-subject and cross-day, respectively.**

***Impact Statement*—Intelligent robot technology has dramatically promoted economic development and social progress. The HCI system is widely implemented in the robotics field because it can fill the gap between human intentions and robotic control, and a more robust generalization HCI system can make the interaction between humans and robots more convenient and efficient. To this end, this study focuses on the transfer learning method to realize the domain adaptation of the sEMG-based hand gesture recognition system, which can improve the generalization of the HCI system. This article first proposes the MKCNN model; then, a transfer learning framework TL-MKCNN based on the proposed MKCNN model is also proposed to improve the performance of the HCI system on different sites.**

***Index Terms*—Domain adaptation, hand gesture recognition, surface electromyogram (sEMG), transfer learning.**

I. Introduction

HUMAN–COMPUTER interaction (HCI) is widely utilized in robotic control for assisting people to interact with surroundings, such as rehabilitation training [1], hand exoskeleton robots [2], and prosthetic limb robots [3]. In an HCI system, physiological, such as the surface electromyogram (sEMG), and kinematical signals are broadly adopted as the interactive information. These signals can fill the gap between the intentions of users and the robotic control [4].

The sEMG-based HCI system can realize the interaction by recognizing different hand gestures with the sEMG signals. For the sEMG-based hand gesture recognition, the conventional machine learning models need to extract and select the handcrafted features by experience [5]–[9]. Phinyomark *et al.* evaluated and selected handcrafted features according to the custom index called the ratio between Euclidean distance and standard deviation [8]. Compared to the conventional machine learning, the deep learning models can extract deep features automatically and obtain better performance on sEMG-based hand gesture recognition [10], [11]. Recent studies have also paid more attention to the sEMG-based hand gesture recognition. For instance, Cote-Allard *et al.* [12] employed a deep learning method to the spectrogram of sEMG signals for recognizing hand gestures. Wei *et al.* [13] decomposed the raw sEMG image into several streams and fused multistream features to recognize gestures.

However, the sEMG-based hand gesture recognition model performs unsatisfactorily on cross-subject and cross-day. Specifically, the model trained by the sEMG signals from some subjects or some days does not perform well on the other subjects or the other days [15], [20]–[22]. Palermo *et al.* [25] constructed a dataset and conducted experiments to verify the repeatability of the hand gesture recognition, and the results show that the accuracy decay problem on cross-subject and cross-day is remarkable. This problem can be regarded as a domain adaptation problem, which means the model trained in one domain cannot be adapted to another domain. To improve the adaptation of the recognition model, the multiclass common spatial patterns with two types of schemes named “one versus one” and “one versus rest” were used for sEMG recognition [26]. Yu *et al.*

Manuscript received November 26, 2020; revised February 22, 2021 and May 2, 2021; accepted July 10, 2021. Date of publication July 26, 2021; date of current version November 17, 2021. This work was supported in part by the National Natural Science Foundation of China under Grant 62025307 and Grant U1913209 and in part by Beijing Municipal Natural Science Foundation under Grant JQ19020. This article was recommended for publication by Associate Editor Catherine Huang upon evaluation of the reviewers’ comments. *(Corresponding author: Long Cheng.)*

The authors are with the State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China, and also with the School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing 100049, China (e-mail: zouyongxiang2019@ia.ac.cn; long.cheng@ia.ac.cn).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TAI.2021.3098253.

Digital Object Identifier 10.1109/TAI.2021.3098253

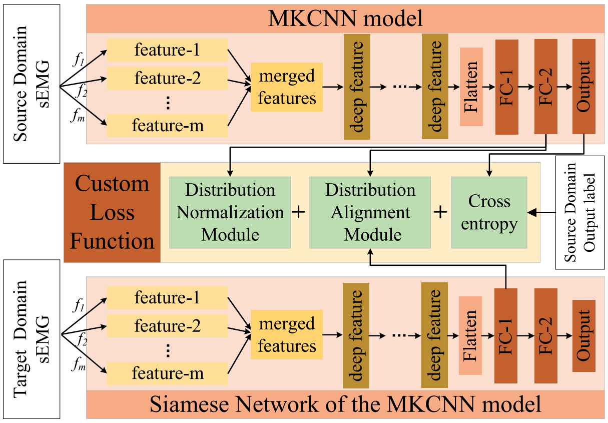


Fig. 1. Framework of the proposed TL-MKCNN.

proposed a deep domain adaptation method for the high-density sEMG-based hand gesture recognition, and based on this work, Wei *et al.* proposed a multiview CNN model for the sparse sEMG-based hand gesture recognition [22].

This study focuses on solving the sEMG-based hand gesture recognition problem on cross-day and cross-subject by transfer learning method to realize domain adaptation. To this end, a multiscale kernel convolutional neural network (MKCNN) model and transfer learning MKCNN (TL-MKCNN) model are proposed. The framework of TL-MKCNN, which is based on MKCNN, is shown in Fig. 1.

The contributions of this article can be concluded as follows.

1) This study proposes an MKCNN model, which uses multiscale convolution kernels to extract multitime-scale features of the multichannel sEMG signals. Additionally, separable convolutional layers, dropout layers, and batch normalization (BN) are adopted to prevent the MKCNN model from overfitting by decreasing the number of training parameters of the model.
2) This study also proposes a transfer learning model based on the proposed MKCNN model, named TL-MKCNN. The TL-MKCNN model is composed of the MKCNN and its Siamese network, and these two parts are connected by the custom loss function, including the custom distribution normalization module (DNM) and the distribution alignment module (DAM). The TL-MKCNN provides a deep learning framework for the sEMG-based hand gesture recognition problem.
3) Experiments are conducted on different public databases to verify the effectiveness of the proposed MKCNN model and TL-MKCNN model. The experimental results show that the proposed TL-MKCNN model can obtain a better performance compared with the state-of-the-art works. On NinaPro DB6, the accuracy of the TL-MKCNN model can achieve 97.22% on within-session, 74.48% on cross-subject, and 90.30% on cross-day.

The remaining parts of this article are organized as follows. Section II introduces the related works, including signal preprocessing methods, the deep learning methods, and the transfer learning methods. Section III states the domain adaptation problem in sEMG-based hand gesture recognition. Section IV describes and explains the structures of the MKCNN model and the TL-MKCNN model, respectively. In Section V, the experiments are conducted on different benchmark databases. Section VI discusses the TL-MKCNN model in the sEMG-base hand gesture recognition problem. Finally, Section VII concludes this article.

II. Related Work

A. Signal Preprocessing

Though the deep learning model can extract features from the raw sEMG signals, the digital filter is necessary to eliminate the interferences of high-frequency random noise, power frequency noise, and the artifact components; additionally, it can also remove the uninteresting bandwidth of sEMG signals. Phinyomark *et al.* adopted the Butterworth finite impulse response filter to obtain the 20–500 Hz frequency band to recognize sEMG-based hand gestures [27]. Upon successful filtering of the raw sEMG signals, the filtered signals need to be segmented before recognizing by a deep learning network. Not only can data segmentation multiplies the number of the samples, but it improves the real-time performance of the model [28]. The adjacent window and the overlapping window are the two most frequently used windowing schemes for segmenting the sEMG signals [29]–[32]. Compared with the adjacent window, the overlapping window has better real-time performance [31], and the time window length between 50–250 ms is suggested [33].

B. Deep Learning Model

Recent studies have demonstrated that deep learning approaches perform well on sEMG-based hand gesture recognition. Chen *et al.* [34] used the recurrent neural network (RNN) model for the skeleton-based dynamic hand gesture recognition, which achieved a good performance. However, the RNN model exists the problem of disaster forgetting [35], so more models based on the long-short term memory (LSTM) are adopted to recognize skeleton-based human activities and hand gestures [36], [37]. Compared to the model based on the RNN framework, the convolutional neural network (CNN) model has a faster calculation speed than RNN and LSTM due to its parallel computing structure. Chen *et al.* [38] employed a compact CNN, which can reduce the number of parameters, to classify hand gestures with the spectrogram of the raw sEMG signals. The high-density sEMG records the spatiotemporal information of the electrical potential from muscle, and every instantaneous multichannel sEMG signal is an array, which can be regarded as an image. The successful applications of the CNN on image provide practical support to recognize high-density sEMG-based hand gestures with the CNN [39]–[41]. Besides, the CNN is also employed to the hand gesture recognition with the sparse channels sEMG signals [22], [42], [43]. However, compared to the images, the information of the raw sEMG signals contains different attributes (temporal and spatial attributes) along the

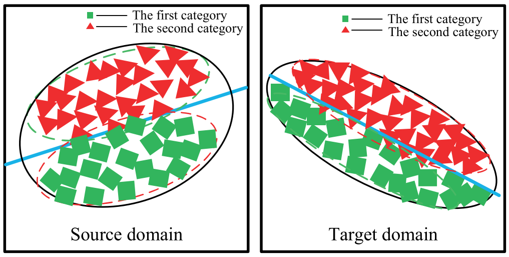


Fig. 2. Distribution difference between source domain and target domain.

length and width direction; hence, the size of the convolutional kernel filter should be further optimized.

C. Transfer Learning

The deep transfer learning method is an effective way to realize domain adaptation [14]. For sEMG-based hand gesture recognition, Cote-Allard *et al.* [15] utilized the sEMG signals from multiple users for training the CNN model to extract the common features of these subjects. Du *et al.* [44] proposed a multistream AdaBN deep domain adaptation method to improve the robustness of the model on cross-subject. The precision decay on cross-subject and cross-day also is remarkable in other physiology-based HCI systems [16]–[18]. He and Wu [19] aligned EEG trials from different subjects in the Euclidean space to make the raw EEG data more similar, and hence improved the learning performance for a new subject. These methods are inspiring for realizing the domain adaptation in the field of sEMG-based hand gesture recognition.

III. Problem Statement

The problem that the sEMG-based hand gesture model does not reach satisfied transfer effectiveness on cross-subject and cross-day is essentially caused by the different distribution characteristics of the train set and the test set, which means the model does not achieve enough generalization on different domains. The domain adaptation of sEMG-based hand gestures can be described as follows. For dataset $\boldsymbol{X_s} \in \mathbb{R}^{m \times n_s}$ with the label of $\boldsymbol{Y_s}$ and $\boldsymbol{X_t} \in \mathbb{R}^{m \times n_t}$ without the label, the feature dimensions of both two dataset are identical to m. Denote $\mathcal{D}_s = \{(\boldsymbol{X_s}, \boldsymbol{Y_s})\}$ as the source domain, and $\mathcal{D}_t = \{\boldsymbol{X_t}\}$ as the target domain, and the source domain and the target domain have similar but not identical distribution, as shown in Fig. 2.

The purpose of transfer learning is to make sure the model trained on the source domain can fit the target domain even when the target domain labels are not available. Specific to this study, it is required that the model has the desired performance on cross-day and cross-subject.

IV. Method

A. Model for Recognition

In this study, an MKCNN is applied to classify the preprocessed sEMG signals. The multichannel sEMG signals can be regarded as an image with the size of $L \times W \times C$, where L is the length of sEMG signals, W is the number of the electrodes of the sEMG acquisition device, and $C = 1$. The length L is determined by the time window t ms and the sample rate of the sEMG signal f kHz ($L = t \times f$). The length and height direction contains the temporal and the spatial information of the sEMG, respectively. The convolution operation can extract the temporal and spatial information of the sEMG simultaneously, and its translation invariance property can enhance the robustness of the model for hand gesture recognition. Besides, some other methods have been taken to improve the generalization of the model and avoid the model overfitting.

First, the number of trainable parameters of the proposed model should be decreased since the redundant parameters of the model are more likely to cause the model overfitting. With the increase of the sample rate f (2000 Hz), the input data size also increases, leading to the increase of the number of the model's parameters. As for this problem, dropout layers are added to every convolution layer, and some conventional convolution layers are replaced by separable convolution layers [45]. The implementation process of a separable convolution layer is composed of two steps: a depthwise convolution operation is performed on the input first, and then a pointwise convolution operation is followed. The task of the depthwise convolution layer is to learn a single convolution kernel corresponding to every channel, resulting in the input and output feature maps of this layer have the same number of channels. The pointwise convolution layers train the 1×1 convolution kernels to output feature maps with the same number of channels as that of input.

Second, BN layers are adopted to improve the generalization of the MKCNN model. In model training, the variational network parameters would also change the distribution of the training data, and this change of distribution is defined as internal covariate shift [46], which can reflect the difference between the training data and test data. Hence, the decrease of the internal covariate shift can improve the generalization of the model. Generally speaking, the BN layer is put before the activation layer and makes the input of the BN layer obeys the normal distribution; therefore, the internal covariate shift can decrease. Besides, because the BN layers whiten the input data, it also accelerates the model training [47]. For m training sEMG samples $X = \{\boldsymbol{x_1}, \boldsymbol{x_2}, \ldots, \boldsymbol{x_m}\}$, BN operation can be expressed as

$$\begin{aligned} \hat{x} &= f(X) \\ x_{\text{std\_norm}} &= \frac{\hat{x} - E[\hat{x}]}{\sqrt{var[\hat{x}] + \epsilon}} \\ \text{BN}_{\gamma,\beta}(X) &= \gamma x_{\text{std\_norm}} + \beta \end{aligned} \tag{1}$$

where $\hat{x}$ is the output of the upper layer neuron, as well as the input of BN layer; $x_{\text{std\_norm}}$ is the standard normalization of $\hat{x}$; γ and β are the translation and scaling parameters the model needs to learn; $\text{BN}_{\gamma,\beta}(X)$ is the output of BN layer.

The aforementioned modules are adopted in the proposed MKCNN model. For different sizes of input, the parameters of MKCNN would accordingly change. For clearly describing the structure of the MKCNN model, assumed that the size of the input sEMG signal is $200 \times 14 \times 1$, which is also the size of the sEMG signal in the NinaPro DB6 after being preprocessed.

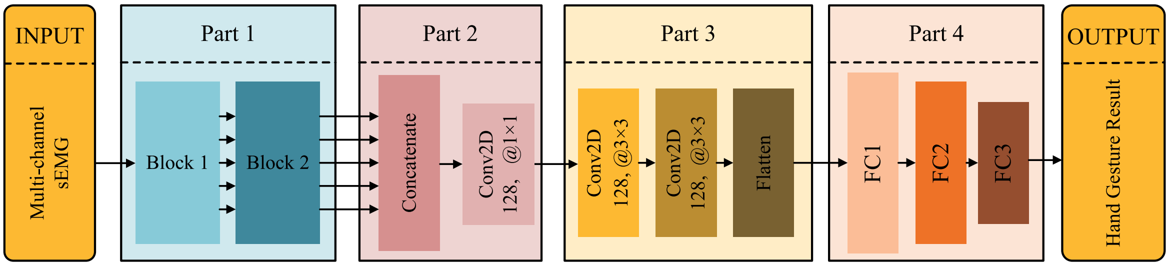


Fig. 3. Architecture of MKCNN model.

TABLE I
STRUCTURES OF PARTS 1–4

Part	Layer Type	Kernel Size	Output Size	Activation	Other Options
Part 1	1–Input	–	$200 \times 14 \times 1$	–	–
	5–Conv2D	$10 \times 3/32$; $20 \times 3/32$;$30 \times 3/32$; $40 \times 3/32$;$50 \times 3/32$	$200 \times n \times 32$	elu	–
	5–BN	–		–	–
	5–MaxPooling2D	10×1	$20 \times 14/32$	–	–
	5–Dropout	–	$20 \times 14/32$	–	rate=0.5
	5–SeparableConv2D	3×3	$20 \times 14/64$	elu	padding = SAME
	5–BN	–	$20 \times 14/64$	–	–
	5–MaxPooling2D	2×2	$10 \times 7/64$	–	–
	5–Dropout	–	$10 \times 7/64$	–	rate=0.5
Part 2	1–Concatenate	–	$10 \times 7/320$	–	–
	1–Conv2D	$1 \times 1/128$	$10 \times 7/128$	elu	–
Part 3	1–SeparableConv2D	3×3	$10 \times 7/128$	elu	padding = SAME
	1–BN	–	$10 \times 7/128$	–	–
	1–MaxPooling2D	2×2	$5 \times 3/128$	–	–
	1–Dropout	–	$5 \times 3/128$	–	rate=0.5
	1–SeparableConv2D	3×3	$5 \times 3/128$	elu	padding = SAME
	1–BN	–	$5 \times 3/128$	–	–
	1–MaxPooling2D	2×2	$5 \times 3/128$	–	–
	1–Dropout	–	$2 \times 1/128$	–	rate=0.5
	1–Flatten	–	256	–	–
Part 4	1–Dense0	–	512	elu	
	1–Dense1	–	128	elu	
	1–Dense2	–	8	softmax	

Note: The number before layer type represents the number of this layer, e.g., 5-Conv2D represents there five conv2D layers to extract the information of its input. When the sample rate decreases, the length of the sEMG sample accordingly decreases. The kernel size of Block 1 of Part 1 should also be modified as $1 \times 3/32$; $2 \times 3/32$; $3 \times 3/32$; $4 \times 3/32$; $5 \times 3/32$.

The architecture of the proposed MKCNN model is presented in Fig. 3. It can be found that MKCNN comprises four parts whose parameters are listed in Table I, and their functions are described as follows.

1) *Part 1*: Two blocks are included in this part, as shown in Fig. 4. This part aims to extract abundant features that can represent the spatiotemporal information of the sEMG signal. Every multichannel sEMG signal sample is regarded as an image. In Block 1, five kinds of filters with the sizes of $10 \times 3, 20 \times 3, 30 \times 3, 40 \times 3$, and 50×3 are employed to extract the temporal and spatial information simultaneously from the input sEMG signals. The sizes of filters affect the receptive field of the neurons, which is related to the temporal information of deep features extracted. The larger the size, the lower the frequency associated with the features. Every output feature map of Block 1 has 32 channels. Followed by Block 2, five identical convolution layers, whose filter size is 3×3, are connected to every convolution layer in Block 1 correspondingly. The separable convolution layers are applied to decrease the number of training parameters of the model, as explained previously. The convolution layers in Block 1 are still traditional convolution layers, and

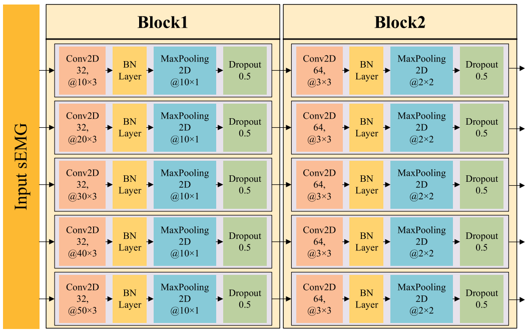


Fig. 4. Structure of the Part 1.

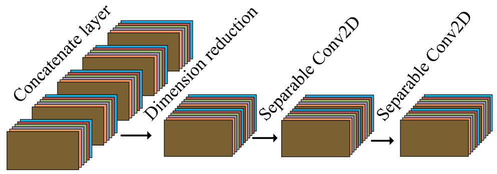


Fig. 5. Structure of the Parts 2 and 3.

the convolution layer structure in Block 2 is separable convolution considering that the input of Block 1 is a single channel image; however, the inputs of Block 2 are multichannel feature maps. Besides, the BN layers and dropout layers are added to these two blocks to improve the generalization of the proposed MKCNN model.

2) *Part 2*: This part aims to reduce the dimension of the output feature maps exported from Part 1 and fuse these feature maps. The filters with the size of 1×1 in this convolution layer are connected after Part 1, making the number of the feature maps decrease from 320 (64×5) to 128.
3) *Part 3*: This part aims to further extract deep features from the fused feature maps by separable convolutional layers. Even though the aforementioned convolutional layers have employed multiscale kernel convolutional layers to extract deep features, deep convolutional layers are still necessary for extracting more semantics. The structure of Parts 2 and 3 is shown in Fig. 5.
4) *Part 4*: The function of this part is to build a mapping relation between deep features and classification results. Two fully connected layers are used to generate more combined deep features and convert these features to the output layer.

Finally, the MKCNN model is built by connecting input, Parts 1–4, and output. The "multiscale kernel" is reflected in Block 1 of Part 1. The hyperparameters of the MKCNN are listed in Table I.

B. Transfer Learning Model

Based on the MKCNN model, the TL-MKCNN model is proposed in this study to solve the problem that the performance of a model decreases on the cross-subject and cross-day. In this study, this problem is regarded as a transfer learning problem. To obtain the desired effect of transfer learning, the TL-MKCNN model needs to extract similar deep features from the sEMG signals belonging to the source domain and the target domain. The distributions of the source domain and the target domain can be used to judge whether the extracted deep features are similar. To this end, a DAM, which uses the Kullback–Leibler divergence to measure the discrepancy between the source domain and the target domain, is designed. The two inputs of the DAM are the deep features of the source domain and the target domain. Note that the distributions of the source domain and the target domain are $p(x)$ and $q(x)$, so the KL divergence between them is

$$\begin{aligned} D_{\mathrm{KL}}(p(x)||q(x)) &= \int p(x) \log \frac{p(x)}{q(x)} dx \\ &= \int p(x) \log p(x) dx - \int p(x) \log q(x) dx \end{aligned} \tag{2}$$

where x represents the deep features in the MKCNN model, it is assumed that the components of variable x are not independent of each other, and both $p(x)$ and $q(x)$ obey Gaussian distribution since x is also the output of the BN layers. The first item in (2) is given by

$$\begin{aligned} &\int p(x) \log p(x) dx \\ &= \int \mathcal{N}(x; \mu_p, \Sigma_p) \log \mathcal{N}(x; \mu_p, \Sigma_p) dx \\ &= -\frac{1}{2} \mathbb{E}_{x \sim p(x)} \left[(x - \mu_p)^T \Sigma_p^{-1} (x - \mu_p) + \log(\det \Sigma_p) \right] \\ &\quad - \frac{J}{2} \log(2\pi) \end{aligned} \tag{3}$$

and the second item in (2) is given by

$$\begin{aligned} &\int p(x) \log q(x) dx \\ &= -\frac{1}{2} \mathbb{E}_{x \sim p(x)} \left[(x - \mu_q)^T \Sigma_q^{-1} (x - \mu_q) + \log(\det \Sigma_q) \right] \\ &\quad - \frac{J}{2} \log(2\pi) \end{aligned} \tag{4}$$

so (2) can be simplified as

$$\begin{aligned} &D_{\mathrm{KL}}(p(x)||q(x)) \\ &= \frac{1}{2} \mathbb{E}_{x \sim p(x)} \left[(x - \mu_q)^T \Sigma_q^{-1} (x - \mu_q) + \log(\det \Sigma_q) \right] \\ &\quad - \frac{1}{2} \mathbb{E}_{x \sim p(x)} \left[(x - \mu_p)^T \Sigma_p^{-1} (x - \mu_p) + \log(\det \Sigma_p) \right] \\ &= \frac{1}{2} \log \frac{\det \Sigma_q}{\det \Sigma_p} + \frac{1}{2} \mathbb{E}_{x \sim p(x)} \left[(x - \mu_q)^T \Sigma_q^{-1} (x - \mu_q) \right] \\ &\quad - \frac{1}{2} \mathbb{E}_{x \sim p(x)} \left[(x - \mu_p)^T \Sigma_p^{-1} (x - \mu_p) \right]. \end{aligned} \tag{5}$$

Since $x \in \mathbb{R}^{J \times 1}$, the result of $(x - \mu)^T \Sigma^{-1} (x - \mu)$ is a scale. It can be simplified as

$$\begin{aligned} &\mathbb{E}_{x \sim p(x)} [(x - \mu)^T \Sigma^{-1} (x - \mu)] \\ &\quad = \mathbb{E}_{x \sim p(x)} [tr((x - \mu)^T \Sigma^{-1} (x - \mu))] \\ &\quad = \mathbb{E}_{x \sim p(x)} [tr(\Sigma^{-1} (x - \mu)(x - \mu)^T)] \\ &\quad = tr(\mathbb{E}_{x \sim p(x)} [\Sigma^{-1} (x - \mu)(x - \mu)^T]) \\ &\quad = \Sigma^{-1} tr(\mathbb{E}_{x \sim p(x)} [(x - \mu)(x - \mu)^T]). \end{aligned} \tag{6}$$

Hence, the third item in (5) can be simplified as

$$\begin{aligned} &\mathbb{E}_{x \sim p(x)} [(x - \mu_p)^T \Sigma_p^{-1} (x - \mu_p)] \\ &\quad = \Sigma_p^{-1} \mathrm{tr}(\mathbb{E}_{x \sim p(x)} [(x - \mu_p)(x - \mu_p)^T]) \\ &\quad = \Sigma_p^{-1} \mathrm{tr}(\Sigma_p) = \mathrm{tr}(I_{N \times N}) = N. \end{aligned} \tag{7}$$

The second item can be simplified as

$$\mathbb{E}_{x \sim p(x)} [(x - \mu_q)^T \Sigma_q^{-1} (x - \mu_q)]$$

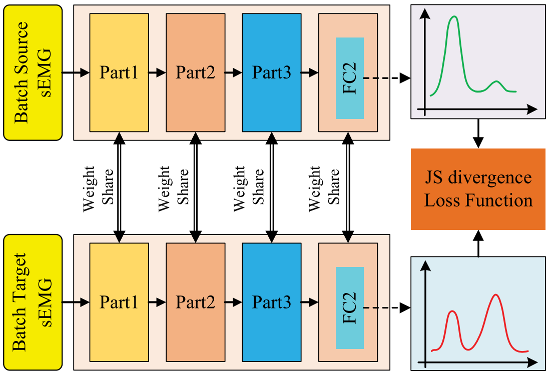


Fig. 6. Structure of DAM.

$$
\begin{aligned}
&= \Sigma_q^{-1}\mathrm{tr}(\mathbb{E}_{x\sim p(x)}[(x-\mu_q)^T(x-\mu_q)])\\
&= \Sigma_q^{-1}\mathrm{tr}(\mathbb{E}_{x\sim p(x)}[x^Tx - x^T\mu_q - \mu_q^Tx + \mu_q^T\mu_q]\\
&= \Sigma_q^{-1}\mathrm{tr}(\mathbb{E}_{x\sim p(x)}(x^Tx - 2\mu_p^T\mu_q + \mu_q^T\mu_q)\\
&= \Sigma_q^{-1}\mathrm{tr}(\Sigma_p + \mu_p^T\mu_p - 2\mu_p^T\mu_q + \mu_q^T\mu_q)\\
&= \Sigma_q^{-1}\mathrm{tr}[\Sigma_p + (\mu_p-\mu_q)^T(\mu_p-\mu_q)]\\
&= \mathrm{tr}(\Sigma_q^{-1}\Sigma_p) + \mathrm{tr}((\mu_p-\mu_q)^T\Sigma_q^{-1}(\mu_p-\mu_q)). \qquad (8)
\end{aligned}
$$

By combining (5), (7), and (8), the KL divergence of two different Gaussian distributions is given by

$$
\begin{aligned}
&D_{\mathrm{KL}}(p(x)||q(x))\\
&\quad= \frac{1}{2}\mathrm{tr}\left[\Sigma_q^{-1}\Sigma_p + (\mu_p-\mu_q)^T\Sigma_q^{-1}(\mu_p-\mu_q)\right]\\
&\qquad+ \frac{1}{2}\log\frac{\det\Sigma_q}{\det\Sigma_p} - \frac{N}{2}. \qquad (9)
\end{aligned}
$$

Even though the KL divergence can measure the difference of two distributions, it does not have the property of symmetry, which means that the exchange of the source domain and the target domain may lead to differences. However, the sEMG signals from different subjects or different days should be equivalent to the model. Therefore, the Jenson–Shannon (JS) divergence is employed to replace the KL divergence to make sure the designed loss function has the property of symmetry, and it is defined as

$$
\begin{aligned}
&D_{\mathrm{JS}}(p(x)||q(x))\\
&\quad= \frac{1}{2}D_{\mathrm{KL}}(p(x)||\frac{p(x)+q(x)}{2})\\
&\qquad+ \frac{1}{2}D_{\mathrm{KL}}(q(x)||\frac{p(x)+q(x)}{2}). \qquad (10)
\end{aligned}
$$

Hence, the DAM can measure the similarity of the distributions of the source domain and the target domain after extracting the deep features of the source domain and the target domain simultaneously. The structure of the DAM module is shown in Fig. 6.

It is worth noting that the dimension of x should not be too high. On the one hand, the batch size during model training is required to be greater than the dimension of x to make sure $\det\Sigma_q$ is not equal to zero. A higher dimension of x would put forward higher requirements for the hardware of the HCI system since the computer memory is consumed in the process of model training. On the other hand, the probability of $p(x)$ and $q(x)$ overlapping in high-dimensional space is small, which means that the value of D_{JS} is very likely to become infinite, leading to the training process becomes meaningless.

It is found that the JS divergence would be infinite in the process of model training, which means that the distributions of the source domain and the target domain are not overlapping. To make these two distributions are overlapped before calculating their JS divergence, a DNM, which adopts a modified center loss function, is designed in the TL-MKCNN model. The modified center loss function can make the deep features of samples from one category as clustered as possible in the high-dimensional space, which increases the overlapping probability of the deep feature from the source domain and the target domain. It is noted that the DNM also has the function of alignment since it aligns the category center points of the deep features from different subjects from the source domain; however, it cannot directly realize the alignment of the distributions from different domains.

To this end, an embedding layer is added to the DNM. Suppose the number of categories of deep features is p, and the dimension of deep features is d. The embedding layer is a $d \times p$ matrix, whose every column stores the center point of the deep features belonging to one category. The function of the DNM is to make the deep features from the same category cluster to its category center points. However, the TL-MKCNN still needs to recognize the sEMG signals of different hand gestures, so the modified center loss function is defined as

$$
\begin{aligned}
L_C &= L_1 + L_2\\
L_1 &= -\log\frac{e^{w_y^\top x_o + b_y}}{\sum_i^n e^{w_i^\top x_o + b_i}}\\
L_2 &= \frac{1}{2}\left\|x - c_y\right\|^2 \qquad (11)
\end{aligned}
$$

where x represents the deep features, $w \in \mathbb{R}^{d\times p}$ is the weight matrix in the last fully connected layer, and w_i denotes the ith column of the weight matrix; $b \in \mathbb{R}^p$ is the bias term, and b_i denotes the ith element of b; c_y denotes the yth column of the output of the embedding layer, and it is worth noting that c is not a constant and should be obtained by training. As a loss function, the first item L_1 represents the cross-entropy loss function of multiclassification, and it trains the model to make samples from the different categories separated as much as possible; the second item L_2 represents the center loss, and it gathers the samples from the same category as much as possible.

After defining the custom loss function, the structure of DNM in this model is shown in Fig. 7.

The TL-MKCNN model needs to combine the DAM, the DNM, the MKCNN model, and the Siamese network of the

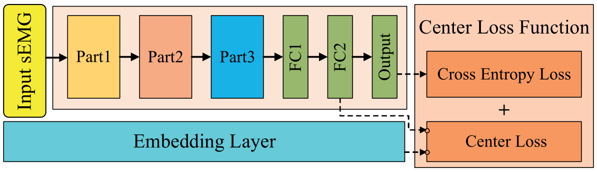


Fig. 7. Structure of DNM.

TABLE II
INFORMATION OF THE NINAPRO DATASET

Name	Number of Gesture	Subjects	Number of Channel	Number of trails	Sample of Rate
DB2 [49]	50	40	12	6	2000Hz
DB3 [49]	50	11	12	6	2000Hz
DB4 [50]	53	10	12	6	2000Hz
DB5 [50]	53	10	16	6	200 Hz
DB6 [25]	8	10	16	120	2000Hz

Note: DB2, DB3, DB4, DB5, and DB6 are the shorthand of the NinaPro DB2, NinaPro DB3, NinaPro DB4, NinaPro DB5, and NinaPro DB6, respectively.

TABLE III
ACCURACY OF MKCNN MODEL AND ITS INCOMPLETE EDITION OF MKCNN MODEL TESTED ON NINAPRO DATABASES

Database	DB2	DB3	DB4	DB6
MKCNN without Dropout	63.73%	67.27%	60.26%	79.84%
MKCNN without BN	70.44%	62.5%	3.01%	86.69%
kernel 1	81.9%	76.6%	72.3%	91.8%
kernel 2	85.4%	73.3%	72.5%	91.5%
kernel 3	84.4%	72.4%	75.0%	91.2%
kernel 4	85.0%	74.5%	73.0%	90.9%
kernel 5	84.9%	68.1%	73.9%	91.6%
MKCNN	**85.7%**	**82.91%**	**76.3%**	**92.91%**

Note: The *kernel 1* represents that the size of convolutional kernel of comparison model in *Block 1*, *Part 1* is 10 × 3. Similarly, kernel 2, kernel 3, kernel 4, and kernel 5 represent that these kernels have the sizes of 20 × 3, 30 × 3, 40 × 3, and 50 × 3, respectively. The bold entities represent the best accuracy of various editions of the MKCNN model on different public data sets.

MKCNN model. Finally, the overall structure of TL-MKCNN is shown in Fig. 8.

V. DATA AND EXPERIMENTS

A. Data Acquisition and Processing

Some experiments are conducted on the public databases to verify the performance of the proposed MKCNN model and the proposed TL-MKCNN model. The NinaPro Project has multiple public databases, which contain the sEMG signals recording different hand gestures. The information of these databases is listed in Table II.

The DB2, DB3, DB4, DB5, and DB6 are used to test the performances of the MKCNN model and TL-MKCNN model on within-session and cross-subject since these databases contain the sEMG signals from different subjects. The DB6 can be used to test the performance of the MKCNN model and TL-MKCNN model on cross-day since it contains the sEMG signals from different days. Consider that the cross-subject and cross-day experiments can be conducted on the DB6, the NinaPro DB6 is mainly introduced in this article.

The NinaPro DB6 is established for studying the repeatability of sEMG signals in gesture recognition. It is recorded from 10 intact subjects with 16 Delsys Trigno sEMG electrodes. All the electrodes are placed around the forearm, and the sEMG signals are sampled at 2000 Hz. The dataset is composed of 7 grasps performed on 14 different objects; every grasp is repeated 12 times, twice a day, for five days. Before data acquisition, the participants are required to sit in front of a table and put their forearms on it. In the process of signal collection, the subjects would complete every hand gesture according to the image shown on the screen. Each gesture lasts for 4 s, and a 4-s rest is followed after every repetition. The collected sEMG signals are filtered with a Hampel filter at 50 Hz.

The third-order Butterworth filter is adopted to process all the sEMG signals, and the 20–500 Hz (50 Hz has been filtered out at the data acquisition stage) band is preserved, and a sliding window with the length of 100 ms and the step size of 25 ms is applied to every trail data to augment the number of training samples. After being preprocessed, the size of every acquired sEMG sample is 200 × n × 1 (sample rate is 2000 Hz) or 20 × n × 1 (sample rate is 200 Hz), and n is the number of electrodes of the sEMG device.

All the samples from all day and subjects are mixed up and shuffled for the within-session experiments, where 80% samples are selected randomly as the train set and validation set to train the model with eight-fold cross-validation, and the rest 20% samples are regarded as the test set. The model is trained on some subjects and tested on the other left subjects for the cross-subject experiments, and trained on some days and tested on the other left days for the cross-day experiments.

B. MKCNN Model Experiments

The comparative experiments on five comparison models are conducted to verify whether the multiscale kernel improves the performance of the MKCNN model. For the comparison models, the sizes of the convolutional kernel in the “Block 1” of “Part 1” are the same; however, the MKCNN model adopts five different sizes of convolutional kernels in the “Block 1” of “Part 1.” The rest structure of the five comparison models is the same as the MKCNN model. Besides, dropout layers and BN layers are adopted in the MKCNN model, whose functions are mentioned in Section IV, and the *MKCNN without dropout layers* model and *MKCNN without BN layers* model are tested on NinaPro databases for verifying whether these layers improve the performance of the MKCNN model. The experimental results are shown in Table III.

As shown in Table III, the multiscale kernel indeed improves the accuracy of the MKCNN model. The results show that the dropout layers and BN layers greatly influence the accuracy of the MKCNN model, and they even affect the convergence of the MKCNN model.

For the MKCNN model, the “elu” is adopted as the activation function of the convolution layers, and the “Adam” optimization method is used to optimize the model. For testing whether the

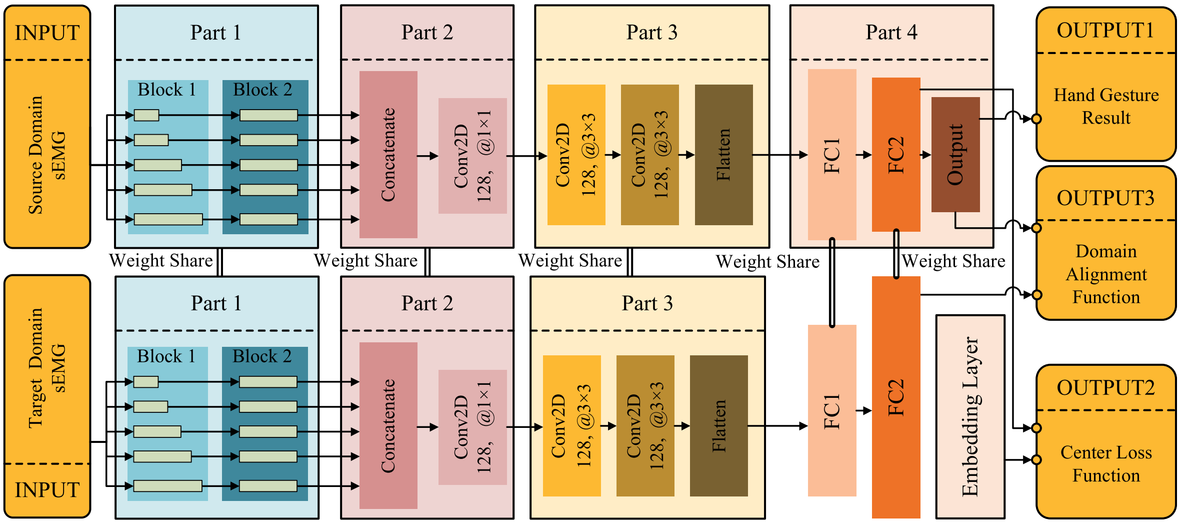


Fig. 8. Structure of TL-MKCNN model.

TABLE IV
ACCURACY OF MKCNN MODEL TESTED WITH DIFFERENT ACTIVATION FUNCTIONS

Activation Function	DB2	DB3	DB4	DB6
elu	**85.70%**	**82.91%**	**76.30%**	**92.91%**
relu	82.36%	80.27%	75.70%	87.23%
tanh	13.36%	11.45%	3.73%	26.26%
sigmoid	83.65%	78.23%	74.26%	87.83%

The bold entities represent the best accuracy of the MKCNN model with different activation functions.

TABLE V
PERFORMANCE OF MKCNN MODEL TESTED WITH DIFFERENT OPTIMIZATION METHODS

Optimization Method	DB2	DB3	DB4	DB6
Adam	**85.70%**	**82.91%**	**76.30%**	**92.91%**
SGD(lr=0.0001)+Momentum	63.73%	60.27%	55.70%	67.76%

The bold entities represent the best accuracy of the MKCNN model based on different optimization methods.

TABLE VI
EFFECTIVENESS OF TRANSFER LEARNING OF THE MKCNN, THE TL-MKCNN, AND THE STATE-OF-THE-ART WORKS TESTED ON NINAPRO DATABASES

Model	Database	Within-session	Cross-subject	Cross-day
AtzoriNet [51]	DB2	60.3 ± 7.7%	–	–
ZhaiNet [52]	DB2	78.71%	–	–
MV-CNN Net [22]	DB2	83.7%	**79.5%**	–
MKCNN	DB2	**85.7%**	63.73%	–
TL-MKCNN	DB2	**86.67%**	65.29%	–
MV-CNN Net [22]	DB3	63.5%	55.5%	–
MKCNN	DB3	**82.91%**	**76.84%**	–
TL-MKCNN	DB3	**87.62%**	**82.47%**	–
MV-CNN Net [22]	DB4	57.50%	54.50%	–
MKCNN	DB4	**76.3%**	**60.10%**	–
TL-MKCNN	DB4	**82.29%**	**62.33%**	–
R-Regularized ELM [25]	DB6	69.83%	41.75%	41.75%
Random Forest [25]	DB6	79.50%	52.43%	52.42%
MV-CNN Net [22]	DB6	64.10%	61.10%	–
MKCNN	DB6	**92.91%**	**62.90%**	**84.79%**
TL-MKCNN	DB6	**97.22%**	**74.48%**	**90.30%**

Note: The key information of DB2, DB3, DB4, and DB6 are listed in Table II.
The bold entities represent the best accuracy of different models on various public databases.

TABLE VII
ACCURACY OF TL-MKCNN MODEL TESTED ON LOW SAMPLE RATE DATABASES (DB5)

Database	TL-MKCNN Accuracy on DB5 (200Hz)	MV-CNN Net [22] Accuracy on DB5 (200Hz)
Within-session	**96.73%**	90.00%
Cross-subject	**83.52%**	83.50%

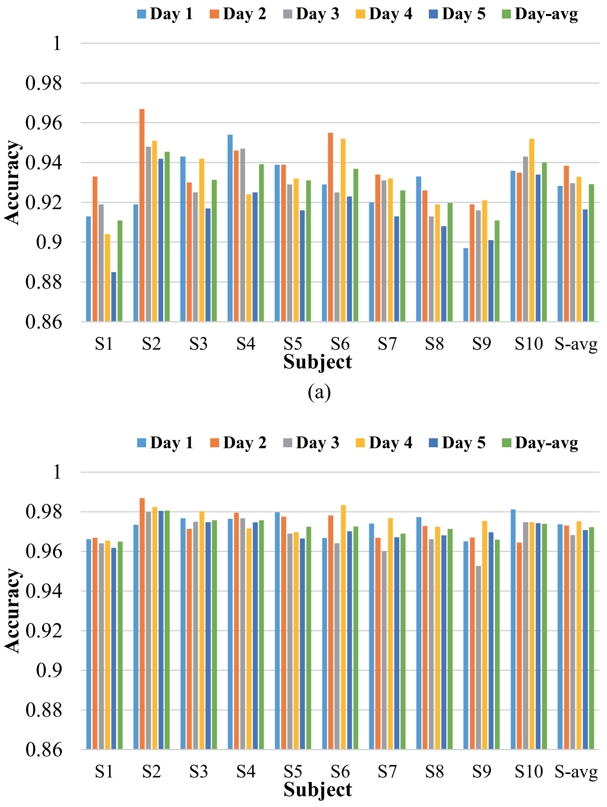
Note: The key information of DB5 is listed in the Table II.
The bold entities represent the best accuracy of different models on low sample rate databases.

choice of the "elu" and the "Adam" is appropriate, the comparison experiments about the selection of activation functions and optimization methods are conducted, whose results are listed in Tables IV and V.

The experimental results show that the "tanh" activation function is likely to lead to the failure of the MKCNN model training, and the "elu" activation function has a better performance in the MKCNN model compared with the cases of the "sigmoid" and the "ReLU" activation functions. As to the optimization methods, the "Adam" can obtain a higher accuracy than "SGD+Momentum."

C. Transfer Learning Model Experiments

The proposed TL-MKCNN model is built based on the MKCNN model. Hence, the experiments of the TL-MKCNN model can be divided into two parts: the first part is the comparison experiments between the MKCNN model and the TL-MKCNN model; the second part is the comparison experiments between the TL-MKCNN model and the state-of-the-art works of the sEMG-based hand gesture recognition.



(a)

(b)

Fig. 9. Classification accuracies of MKCNN and TL-MKCNN models tested on within-session. (a) Test result of MKCNN. (b) Test result of TL-MKCNN. The Day-avg and S-avg represent the average of different days and different subjects, respectively. Every Si ($i \in 1, 2, \ldots, 10$) represents a subject and "S-avg" represents the average result of S1 to S10. Every Si ($i \in 1, 2, \ldots, 10$) contains six items; the first five items represent the accuracy of the model tested on the same Si subject but on different days (day 1, day 2, . . ., day 5), the sixth item represents the average accuracy of this Si subject on these five days. The "Day-avg" item in "S-avg" can reflect the overall average classification accuracy on within-session.

In the process of model training and prediction, the labeled train dataset is regarded as the source domain, and the unlabeled test dataset is regarded as the target domain. The proposed TL-MKCNN model can be trained with only the labeled samples from the source domain. In this training case, the custom loss function of the TL-MKCNN only contains the DNM and cross-entropy, and the structure of the TL-MKCNN can be simplified, as shown in Fig. 7. For the recognition on within-session, the TL-MKCNN is trained in this manner since the train set and test set come from the same domain. The experiments have been conducted on the DB6 to compare the performance of the MKCNN and the TL-MKCNN on within-session. The experimental results are shown in Fig. 9.

The experimental results in Fig. 9 show that the average accuracy of TL-MKCNN can reach 97.22% on within-session, which is 4.31% higher than that of the MKCNN model. Although the main purpose of TL-MKCNN is to improve the generalization of the MKCNN on cross-subject and cross-day, the results show that the TL-MKCNN also improves the performance of the TL-MKCNN model on within-session.

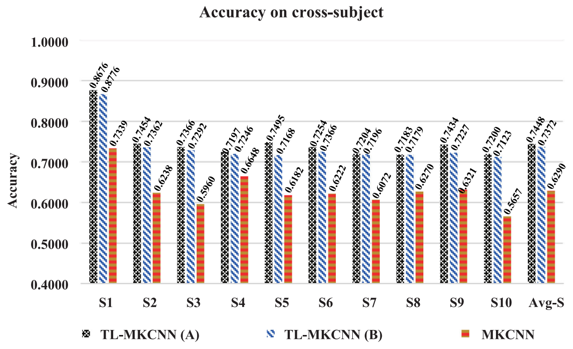


Fig. 10. Classification accuracies of MKCNN and TL-MKCNN model tested on cross-subject. The TL-MKCNN (A) represents the TL-MKCNN trained with only the labeled data from the source domain, and the TL-MKCNN (B) represent the TL-MKCNN trained with both the labeled data from the source domain and unlabeled data from the target domain.

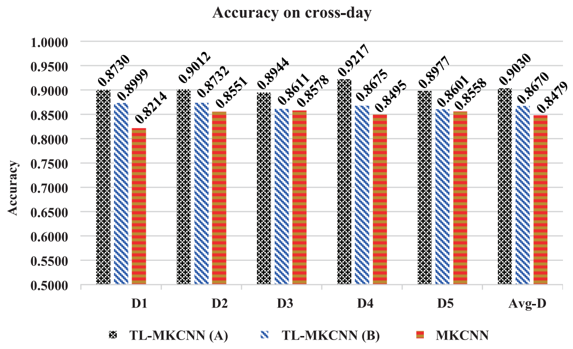


Fig. 11. Classification accuracies of MKCNN and TL-MKCNN model tested on cross-day. The TL-MKCNN (A) represents the TL-MKCNN trained with only the labeled data from the source domain, and the TL-MKCNN (B) represents the TL-MKCNN trained with both the labeled data from the source domain and unlabeled data from the target domain.

The TL-MKCNN can also be trained with both the labeled data of the source domain and unlabeled data of the target domain. When the TL-MKCNN predicts the hand gestures of the subjects from the target domain (cross-subject and cross-day), the labels of these sEMG signals are unknown, which are the future instances for the testing process. However, these samples are not necessary the samples of the prediction moment; hence, they are available for the training process. Consider that these unlabeled sEMG samples from the target domain can imply the distribution of the sEMG samples from the target domain, so these samples should be fully utilized to train the TL-MKCNN. In this training case, the custom loss function of the TL-MKCNN contains the DNM, the DAM, and the cross-entropy items, and the structure of the TL-MKCNN is shown in Fig. 8. The experiments in which the MKCNN and the TL-MKCNN are trained with or without the unlabeled data of the target domain have been conducted on the DB6. The experimental results are shown in Figs. 10 and 11, respectively.

Experimental results show that the accuracies of the TL-MKCNN model can obtain 73.72% on cross-subject and 86.70% on cross-day when the model is trained with only the labeled samples from the source domain, which is 10.83% and 1.91% higher than that of the MKCNN model on cross-subject and cross-day, respectively; the TL-MKCNN model can obtain 73.72% accuracy on cross-subject and 86.70% accuracy on cross-day when the model is trained with both the labeled data of the source domain and unlabeled data of the target domain, which is 11.58% and 5.51% higher than that of the MKCNN model on cross-subject and cross-day, respectively. In both training cases, the TL-MKCNN can obtain better performance compared to the state-of-the-art works.

Although the training needs the unlabeled samples from the target domain, it has no interference to the users at all. From the perspective of the users' convenience, no matter the train dataset contains the unlabeled samples from the target domain or not, and the training process has no differences to the users when using this HCI system; however, the training with unlabeled samples from the target domain can further improve 0.76% and 3.60% accuracies on cross-subject and cross-day, respectively. Therefore, the train dataset comprises both the labeled data of the source domain and unlabeled data of the target domain.

The experimental results are desirable and reasonable. In the first training case (only sEMG data from the source domain), the TL-MKCNN requires the deep features extracted from different domains to be clustered to the category center points in the feature space, leading to these deep features extracted by the model are the common deep features of subjects from different domains so that it can be realized domain adaptation. In the second training case (with both sEMG samples from the source domain and sEMG samples from the target domain), the TL-MKCNN further aligns the overall distribution of deep features except for aligning their category center points of these deep features. The information of the unlabeled samples from the target domain improves the performance of the TL-MKCNN model.

The TL-MKCNN model is built by adding the DAM and the DNM to the MKCNN model. To intuitively present the difference between the MKCNN model and the TL-MKCNN model, the t-SNE method is used to visualize the high-dimensional deep features of the full connection layer (the second full connection layer in Part 4 of TL-MKCNN). The comparison result is shown in Fig. 12, which shows that both the MKCNN and TL-MKCNN can extract clustered deep features, and the deep features extracted by the TL-MKCNN model have a more robust aggregation.

More experiments are conducted on DB2, DB3, DB4, and DB6 to compare the TL-MKCNN model and MKCNN model with the state-of-the-art works. These databases do not contain the labels about days. Hence, only the within-session and cross-subject experiments can be conducted. The experimental results are listed in Table VI.

The experimental results show that the TL-MKCNN model has a better performance compared to the state-of-the-art works. On the DB6, the average accuracy of TL-MKCNN is 17.72% higher than that of "MV-CNN Net" on within-session, 13.38% higher than that of "MV-CNN Net" on cross-subject, and 37.88% higher than that of "Random Forest" on cross-day.

It is noted that all the aforementioned experiments are conducted on high sample rate sEMG signals. The performance of the TL-MKCNN model on low sample rate is also tested with NinaPro DB5, whose sEMG signals are collected with the sample rate of 200 Hz. The experimental results are list in Table VII.

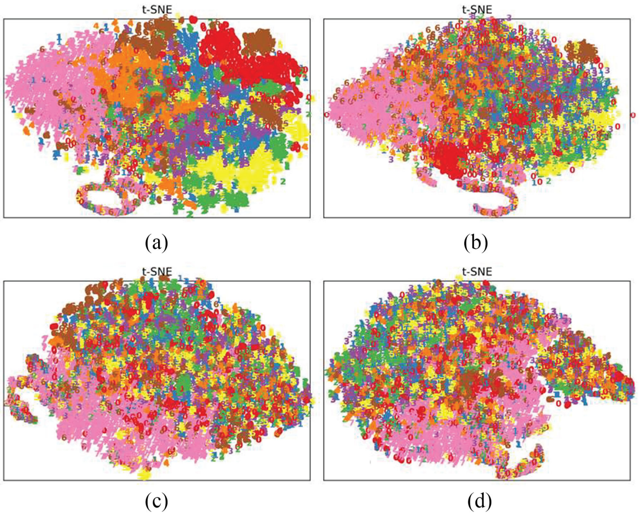


Fig. 12. Visual results of deep features extracted by MKCNN and TL-MKCNN on cross-day and cross-subject. (a) Deep features extracted by the TL-MKCNN model on cross-day. (b) Deep features extracted by the TL-MKCNN model on cross-subject. (c) Deep features extracted by the MKCNN model on cross-day. (b) Deep features extracted by the MKCNN model on cross-subject.

The experimental results show that the TL-MKCNN model can also be applied to the low sample rate sEMG signals. It can obtain the desired performance compared with the state-of-the-art works on DB5. The average accuracy of TL-MKCNN is 6.73% higher than that of "MV-CNN Net" on within-session.

VI. Discussion

This article focuses on the sEMG-based hand gesture recognition, which is booming in HCI. However, this recognition system faces the problem of accuracy decay on cross-subject and cross-day, which is caused by the individual difference, the change of the status of users, and the change of positions of electrodes while wearing the sEMG acquisition device. A transfer learning method is adopted to solve this problem. The MKCNN model is proposed first; then, the TL-MKCNN model is built to improve the generalization of the model on cross-subject and cross-day by applying a transfer learning framework based on the MKCNN model. The experimental results show that the TL-MKCNN model can obtain the desired performance on cross-subject and cross-day. The enhancement of generalization on cross-subject and cross-day can further promote the development of HCI systems, which also positively impacts the development of intelligent robots, such as hand exoskeleton robots, prosthetic limb robots, and intelligent wearable robots. This study also has some limitations. It focuses on the sparse channel sEMG signals; however, the high-density sEMG signal recognition is not discussed in this article.

VII. Conclusion

This article aims to propose an sEMG-based hand gestures recognition model and obtain the desired performance on cross-subject and cross-day. To this end, this article first proposes an MKCNN model with multiscale convolution kernels, and the convolution kernels of different sizes we used to process the input sEMG signal simultaneously, which can obtain multitime scale deep features. Based on the MKCNN model, the TL-MKCNN model is proposed. The custom loss function was used in the TL-MKCNN model to learn the same or similar distribution characteristics from different subjects or the same subject but on different days. Some experiments we conducted on the Ninapro DB2, Ninapro DB3, Ninapro DB4, Ninapro DB5, and Ninapro DB6 to verify the performance of the proposed MKCNN and TL-MKCNN models. The experimental results showed that the MKCNN model and the TL-MKCNN model can obtain the desired performance. On the DB6, the average accuracy of TL-MKCNN is 17.72% higher than that of “MV-CNN NET” [22] on within-session, 7.03% higher than that of “MV-CNN NET” [22] on cross-subject, and 37.88% higher than that of “Random Forest” [25] on cross-day. High-density sEMG signals will be researched in future work, providing a good foundation for HCI or health detection and evaluation problems.

References

[1] M. Xing, G. Wei, J. Liu, J. Zhang, F. Yang, and H. Cao, “A review on multi-modal human motion representation recognition and its application in orthopedic rehabilitation training,” *J. Biomed. Eng.*, vol. 37, no. 1, pp. 174–178, 2020.

[2] Y. Zhuang, Y. Leng, J. Zhou, R. Song, L. Li, and S. W. Su, “Voluntary control of an ankle joint exoskeleton by able-bodied individuals and stroke survivors using EMG-based admittance control scheme,” *IEEE Trans. Biomed. Eng.*, vol. 68, no. 2, pp. 695–705, Feb. 2021.

[3] J. Lázaro, N. Reljin, M.-B. Hossain, Y. Noh, P. Laguna, and K. H. Chon, “Wearable armband device for daily life electrocardiogram monitoring,” *IEEE Trans. Biomed. Eng.*, vol. 67, no. 12, pp. 3464–3473, Dec. 2020.

[4] H. Liu *et al.*, “An epidermal sEMG tattoo-like patch as a new human-machine interface for patients with loss of voice, ” *Microsyst Nanoeng.*, vol. 6, pp. 1–13, 2020.

[5] N. Jose, R. Raj, P. K. Adithya, and K. S. Sivanadan, “Classification of forearm movements from sEMG time domain features using machine learning algorithms,” in *Proc. IEEE Region 10 Conf.*, Penang, Malaysia, 2017, pp. 1624–1628.

[6] D. Karabulut, F. Ortes, Y. Z. Arslan, and M. A. Adli, “Comparative evaluation of EMG signal features for myoelectric controlled human arm prosthetics,” *Biocybernetics Biomed. Eng.*, vol. 37, no. 2, pp. 326–335, 2017.

[7] M. Jochumsen, A. Waris, and E. N. Kamavuako, “The effect of arm position on classification of hand gestures with intramuscular EMG,” *Biomed. Signal Process. Control*, vol. 43, pp. 1–8, 2018.

[8] A. Phinyomark, S. Hirunviriya, C. Limsakul, and P. Phukpattaranont, “Evaluation of EMG feature extraction for hand movement recognition based on Euclidean distance and standard deviation,” in *Proc. ECTI Int. Conf. Elect. Eng./Electron., Comput., Telecommun. Inf. Technol.*, Chiang Mai, Thailand, May 2010, pp. 856–860.

[9] C. N. Savithri and E. Priya, “Correlation coefficient based feature selection for actuating myoelectric prosthetic arm,” in *Proc. Trends Ind. Meas. Autom.*, Chennai, India, 2017.

[10] 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, 2018, Art. no. e0206049.

[11] Y. Zhang, Y. Chen, H. Yu, X. Yang, and W. Lu, “Learning effective spatial-temporal features for sEMG armband-based gesture recognition,” *IEEE Internet Things J.*, vol. 7, no. 8, pp. 6979–6992, Aug. 2020.

[12] U. Cote-Allard *et al.*, “A convolutional neural network for robotic arm guidance using sEMG based frequency-features,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Daejeon, South Korea, 2016, pp. 2464–2470.

[13] W. Wei, Y. Wong, Y. Du, Y. Hu, M. S. Kankanhalli, and W. Geng, “A multi-stream convolutional neural network for sEMG-based gesture recognition in muscle-computer interface,” *Pattern Recognit. Lett.*, vol. 119, pp. 131–138, 2017.

[14] S. Niu, Y. Liu, J. Wang, and H. Song, “A decade survey of transfer learning, (20 10–2020),” *IEEE Trans. Artif. Intell.*, vol. 1, no. 2, pp. 151–166, Oct. 2020.

[15] U. Cote-Allard *et al.*, “Deep learning for electromyographic hand gesture signal classification using transfer learning,” *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 27, no. 4, pp. 760–771, Apr. 2019.

[16] Y.-P. Lin, “Constructing a personalized cross-day EEG-based emotion-classification model using transfer learning,” *IEEE J. Biomed. Health Informat.*, vol. 24, no. 5, pp. 1255–1264, May 2020.

[17] W. Hang *et al.*, “Cross-subject EEG signal recognition using deep domain adaptation network,” *IEEE Access*, vol. 7, pp. 128273–128282, 2019.

[18] H. He and D. Wu, “Different set domain adaptation for brain-computer interfaces: A label alignment approach,” *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 28, no. 5, pp. 1091–1108, May 2020.

[19] H. He and D. Wu, “Transfer learning for brain–computer interfaces: A Euclidean space data alignment approach,” *IEEE Trans. Biomed. Eng.*, vol. 67, no. 2, pp. 399–410, Feb. 2020.

[20] X. Chen, Y. Li, R. Hu, X. Zhang, and X. Chen, “Hand gesture recognition based on surface electromyography using convolutional neural network with transfer learning method,” *IEEE J. Biomed. Health Informat.*, vol. 25, no. 4, pp. 1292–1304, Apr. 2021.

[21] M. Z. ur Rehman *et al.*, “Stacked sparse autoencoders for EMG-based classification of hand motions: A comparative multi day analyses between surface and intramuscular EMG, ” *Appl. Sci.*, vol. 8, no. 7, pp. 11–26, 2018.

[22] 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.*, vol. 66, no. 10, pp. 2964–2973, Oct. 2019.

[23] W. Wu, Z. Chen, X. Gao, Y. Li, E. N. Brown, and S. Gao, “Probabilistic common spatial patterns for multichannel EEG analysis,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 3, pp. 639–653, Mar. 2015.

[24] M. Dai, D. Zheng, S. Liu, and P. Zhang, “Transfer kernel common spatial patterns for motor imagery brain-computer interface classification,” *Comput. Math. Methods Med.*, vol. 2018, 2018, Art. no. 9871603.

[25] F. Palermo, M. Cognolato, A. Gijsberts, H. Muller, B. Caputo, and M. Atzori, “Repeatability of grasp recognition for robotic hand prosthesis control based on sEMG data,” in *Proc. Int. Conf. Rehabil. Robot.*, vol. 2017, pp. 1154–1159, Jul. 2017.

[26] 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, 2015, Art. no. 110.

[27] A. Phinyomark, R. N. Khushaba, and E. Scheme, “Feature extraction and selection for myoelectric control based on wearable EMG sensors,” *Sensors*, vol. 18, no. 5, 2018, Art. no. 1615.

[28] O. W. Samuel *et al.*, “Intelligent EMG pattern recognition control method for upper-limb multifunctional prostheses: Advances, current challenges, and future prospects,” *IEEE Access*, vol. 7, pp. 10150–10165, 2019.

[29] T. R. Farrell, “Determining delay created by multifunctional prosthesis controllers,” *J. Rehabil. Res. Develop.*, vol. 48, no. 6, pp. 21–38, 2011.

[30] M. A. Oskoei and H. Hu, “Myoelectric control systems—A survey,” *Biomed. Signal Process. Control*, vol. 2, no. 4, pp. 275–294, 2007.

[31] M. A. Oskoei and H. Hu, “Support vector machine-based classification scheme for myoelectric control applied to upper limb,” *IEEE Trans. Biomed. Eng.*, vol. 55, no. 8, pp. 1956–1965, Aug. 2008.

[32] K. Englehart and B. Hudgins, “A robust, real-time control scheme for multifunction myoelectric control,” *IEEE Trans. Biomed. Eng.*, vol. 50, no. 7, pp. 848–854, Jul. 2003.

[33] L. H. Smith, L. J. Hargrove, B. A. Lock, and T. A. Kuiken, “Determining the optimal window length for pattern recognition-based myoelectric control: Balancing the competing effects of classification error and controller delay,” *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 19, no. 2, pp. 186–192, Apr. 2011.

[34] X. Chen, H. Guo, G. Wang, and L. Zhang, “Motion feature augmented recurrent neural network for skeleton-based dynamic hand gesture recognition,” in *Proc. IEEE Int. Conf. Image Process.*, Beijing, China, 2017, pp. 2881–2885.

[35] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[36] J. C. Nez, R. Cabido, J. J. Pantrigo, A. S. Montemayor, and J. F. Vlez, “Convolutional neural networks and long short-term memory for skeleton-based human activity and hand gesture recognition,” *Pattern Recognit.*, vol. 76, pp. 80–94, 2018.

[37] R. N. Khushaba, A. Phinyomark, A. H. Al-Timemy, and E. Scheme, “Recursive multi-signal temporal fusions with attention mechanism improves EMG feature extraction,” *IEEE Trans. Artif. Intell.*, vol. 1, no. 2, pp. 139–150, Oct. 2020.

[38] L. Chen, J. Fu, Y. Wu, H. Li, and B. Zheng, “Hand gesture recognition using compact CNN via surface electromyography signals,” *Sensors*, vol. 20, no. 3, 2020, Art. no. 672.

[39] W. Geng, Y. Du, W. Jin, W. Wei, Y. Hu, and J. Li, “Gesture recognition by instantaneous surface EMG images,” *Sci. Rep.*, vol. 6, 2016, Art. no. 36571.

[40] M. R. Islam, D. Massicotte, and W. Zhu, “All-ConvNet: A lightweight all CNN for neuromuscular activity recognition using instantaneous high-density surface EMG images,” in *Proc. IEEE Int. Instrum. Meas. Technol. Conf.*, Dubrovnik, Croatia, 2020, pp. 1–6.

[41] S. Tam, M. Boukadoum, A. Campeau-Lecours, and B. Gosselin, “A fully embedded adaptive real-time hand gesture classifier leveraging HD-sEMG and deep learning,” *IEEE Trans. Biomed. Circuits Syst.*, vol. 14, no. 2, pp. 232–243, Apr. 2020.

[42] U. Cote-Allard, C. L. Fall, A. Campeau-Lecours, C. Gosselin, F. Laviolette, and B. Gosselin, “Transfer learning for sEMG hand gestures recognition using convolutional neural networks,” in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, 2017, pp. 1663–1668.

[43] Y. Li, X. Chen, X. Zhang, K. Wang, and Z. J. Wang, “A sign-component-based framework for Chinese sign language recognition using accelerometer and sEMG data,” *IEEE Trans. Biomed. Eng.*, vol. 59, no. 10, pp. 2695–2704, Oct. 2012.

[44] 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, Mar. 2017, Art. no. 458.

[45] F. Chollet, “Xception: Deep learning with depthwise separable convolutions,” in *Proc. 30th IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 1800–1807.

[46] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” in *Proc. 32nd Int. Conf. Mach. Learn.*, Lille, France, 2015, pp. 448–456.

[47] S. Wiesler and H. Ney, “A convergence analysis of log-linear training,” in *Proc. 24th Int. Conf. Neural Inf. Process. Syst.*, Granada, Spain, 2010, pp. 657–665.

[48] M. Atzori *et al.*, “Electromyography data for non-invasive naturally-controlled robotic hand prostheses,” *Sci. Data*, vol. 1, no. 1, 2014, Art. no. 140053.

[49] S. Pizzolato, L. Tagliapietra, M. Cognolato, M. Reggiani, H. Muller, and M. Atzori, “Comparison of six electromyography acquisition setups on hand movement classification tasks,” *PLoS ONE*, vol. 12, no. 10, 2017, Art. no. e0186132.

[50] 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 Neurorobot.*, vol. 10, pp. 9–18, 2016.

[51] X. Zhai, B. Jelfs, R. H. M. Chan, and C. Tin, “Self-recalibrating surface EMG pattern recognition for neuroprosthesis control based on convolutional neural network,” *Frontiers Neurosci.*, vol. 11, pp. 379–389, 2017.

Yongxiang Zou received the B.E. degree in mechanical engineering and automation and the M.S. degree in mechanical and electronic engineering from the Wuhan University of Technology, Wuhan, China, in 2016 and 2019, respectively. He is currently working toward the Ph.D. degree in control theory and control engineering with the State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing, China.

He is currently with the School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing, China. His current research interests include intelligent rehabilitation assessment, multimodal information fusion, and human–computer interaction system.

Long Cheng (Senior Member, IEEE) received the B.S. (Hons.) degree in control engineering from Nankai University, Tianjin, China, in 2004, and the Ph.D. (Hons.) degree in control theory and control engineering from the Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 2009.

He is currently a Full Professor with the Institute of Automation, Chinese Academy of Sciences. He is also an Adjunct Professor with the University of Chinese Academy of Sciences, Beijing, China. He has authored or coauthored more than 100 technical papers in peer-refereed journals and prestigious conference proceedings. His current research interests include the rehabilitation robot, intelligent control, and neural networks.

Dr. Cheng was a recipient of the IEEE TRANSACTIONS ON NEURAL NETWORKS Outstanding Paper Award from the IEEE Computational Intelligence Society, the Aharon Katzir Young Investigator Award from the International Neural Networks Society, and the Young Researcher Award from the Asian Pacific Neural Networks Society. He is currently serving as an Associate Editor/Editorial Board Member for the IEEE TRANSACTIONS ON CYBERNETICS, *Neural Processing Letters*, *Neurocomputing*, *International Journal of Systems Science*, and *Acta Automatica Sinica*.