Smart Health 23 (2022) 100229

Contents lists available at ScienceDirect

Smart Health

journal homepage: www.elsevier.com/locate/smhl

Privacy computing using deep compression learning techniques for neural decoding

Huining Li [a,*], Huan Chen [b], Chenhan Xu [a], Anarghya Das [a], Xingyu Chen [c], Zhengxiong Li [c], Jian Xiao [d], Ming-Chun Huang [e], Wenyao Xu [a]

[a] Department of Computer Science and Engineering, University at Buffalo, United States
[b] Department of Electrical, Computer, and Systems Engineering, Case Western Reserve University, United States
[c] Department of Computer Science and Engineering, University of Colorado Denver, United States
[d] School of Electronic and Control Engineering, Chang'an University, China
[e] Department of Data and Computational Science, Duke Kunshan University, China

ARTICLE INFO

Keywords:
Brain–computer interface
Privacy computing
Neural decoding
Unsupervised deep learning

ABSTRACT

The brain–computer interface supports a variety of applications with the help of machine learning technology. However, existing edge-cloud infrastructure requires subjects to send their sensitive neural signals to the cloud for training the model which brings privacy concerns. Although the recent distributed learning technology is used to help protect subjects' privacy, it brings high communication costs and cannot avoid privacy reconstruction attacks. In this paper, we propose deep compression learning techniques in the privacy computing infrastructure that can be used for neural decoding while preserving privacy and largely reducing the communication cost. Specifically, we first perform heterogeneous neural signals processing and convert them to resized functional brain connectivity images. Then, a semantics structure-based unsupervised deep compression learning network is trained and generates a neural hash locally for each image. Each hash value is irreversible that cannot be used to reconstruct the user's original neural signal. After that, the cloud end receives the uploaded neural hashes and corresponding labels and filters the abnormal ones. Finally, these neural hashes are used for training neural decoding models. Since a single hash value can correspond to different types of labels, it only needs to be uploaded once with a very small size and then reused for different training tasks, which largely reduce communication cost. Our experiment results show that the proposed privacy computing techniques can be applied to heterogeneous neural signals for training different neural decoding models where the relative accuracy can achieve above 83%.

1. Introduction

A brain–computer interface (BCI) system can provide a high-throughput communication method to convey brain messages independent from the brain's normal output pathway (Santhanam, Ryu, Byron, Afshar, & Shenoy, 2006). With the development of machine learning technology, BCI becomes increasingly promising and popular for human–machine interaction, which can support a wide variety of applications including command inference, emotion recognition, disease detection, etc (Wolpaw et al., 2000).

* Corresponding author.
E-mail addresses: huiningl@buffalo.edu (H. Li), hxc556@case.edu (H. Chen), chenhanx@buffalo.edu (C. Xu), anarghya@buffalo.edu (A. Das), XINGYU.CHEN@UCDENVER.EDU (X. Chen), ZHENGXIONG.LI@UCDENVER.EDU (Z. Li), xiaojian@chd.edu.cn (J. Xiao), mh596@duke.edu (M.-C. Huang), wenyaoxu@buffalo.edu (W. Xu).

https://doi.org/10.1016/j.smhl.2021.100229
Received 24 October 2021; Accepted 5 November 2021
Available online 27 November 2021

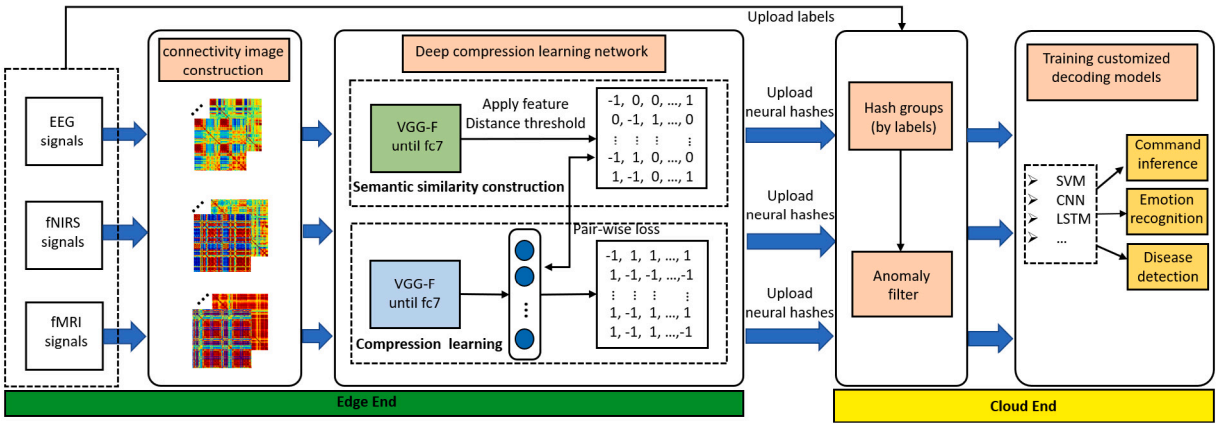

Fig. 1. A privacy computing infrastructure using deep compression learning techniques for training neural decoding models in edge-cloud brain computer interface.

However, traditional edge-cloud infrastructure and machine learning techniques are required to send subjects' sensitive brain data from the edge to the cloud for training, which brings concerns in security and privacy.

Existing works explored some distributed deep learning-based neural decoding approaches in BCI to protect users' privacy (Bercea, Wiestler, Rueckert, & Albarqouni, 2021; Wu, Chen, Zhou, & Zhang, 2020). In distributed deep learning, each party trains a model using private data locally and only uploads the model parameters to the cloud for updating the global model. Gao, Ju, Wei, Liu, Chen, and Yang (2019) developed a heterogeneous federated learning model for training over heterogeneous EEG data. Li et al. (2020) proposed a privacy-preserving federated learning framework that adds Gaussian and Laplace noise into local model weights for training the fMRI-based decoding models. To summarize, these existing works mainly have two limitations: 1) *High communication cost*: Distributed learning is typically limited by an upload bandwidth of 1 MB/s or less (McMahan, Moore, Ramage, Hampson, & y Arcas, 2017), and needs many communication rounds for convergence, therefore communication is a critical bottleneck. 2) *Privacy recovery*: User private data (e.g., age, gender, health condition) is likely to be recovered from uploaded model weights and gradients in distributed deep learning (Melis, Song, De Cristofaro, & Shmatikov, 2019). Although adding Gaussian and Laplace noise may defense reconstruction attacks, it will increase iteration times to converge and bring more communication costs. Therefore, how to preserve user's privacy and save communication costs are still an ongoing issue in training recognition models for BCI applications.

To this end, we develop deep compression learning techniques in the privacy computing infrastructure for neural decoding. To implement our framework, we first pre-process and convert brain signals (e.g., EEG, fNIRS, fMRI) to brain functional connectivity images, and then a semantics structure-based unsupervised deep compression learning network is trained based on these images for generating neural hashes in the edge. Hash values are irreversible that cannot be used to reconstruct a user's privacy. After that, these neural hashes are uploaded to the cloud and grouped by labels, and a statistical autonomous and data-driven approach is leveraged to filter abnormal hash values. Finally, these neural hashes are used for training neural decoding models centralized. Since a single neural hash can correspond to different types of labels, it only needs to be uploaded once with a very small size and reused for different training tasks, which largely reduces communication costs.

Our contributions are summarized as three-fold:

- We develop a distributed deep compression learning framework in the privacy computing infrastructure that preserves users' privacy and largely reduces communication cost in training neural decoding models to support extensive BCI applications.
- We implement functional brain connectivity image construction, deep compression learning network, and anomaly filter to deal with heterogeneous EEG data, fNIRS data, and fMRI data in our framework.
- We leverage relative accuracy to evaluate the neural decoding performance with different hash sizes in three applications.

2. Framework overview

Edge end: In the edge end, collected brain signals are first pre-processed and converted into functional brain connectivity images. Then, each participant trains a semantic structure-based unsupervised deep compression learning network based on a pre-trained model locally and generates neural hashes, which are transmitted to the cloud server along with labels afterward, as shown in Fig. 1.
Cloud end: In the cloud server, the received neural hashes from different participants are grouped by their labels, and then an abnormal neural hash filter is applied to find and discard the inconsistent hash values. These uploaded neural hashes can be reused in the training process for developing customized neural decoding models for different applications.

3. Methodology

3.1. Heterogeneous neural signal processing

The fMRI sensor collects a sequence of whole brain volumes data in a time interval. In fMRI scan, a voxel is the unit of the brain volume. Therefore, raw fMRI data is four-dimensional, i.e., three dimensions represent spatial information and one dimension represents temporal information. After realignment, slice time correction, normalization, smoothing, Blood Oxygenation Level Dependent (BOLD) time series of each voxel is calculated by using a filtering approach (Athanasiou, 2014; Yan, Wang, Zuo, & Zang, 2016). BOLD time series represents the change of activation of the neurons over a time period.

For fNIRS measurement, time sequences of optical intensity changes from different channels are recorded. Then, we apply a bandpass filter of 0.01 to 0.3 Hz to mitigate the physiological noise caused by heartbeat and respiration. After that, time series of relative changes in concentrations of oxygenated hemoglobin (ΔHBO) for each channel is calculated based on the modified Beer–Lambert law (Hoshi, 2003).

For EEG signals, we first use filter methods to remove extrinsic artifacts such as sweating and drift in the electrode impedance, and then apply independent component analysis to remove some intrinsic artifacts caused by eye movement, blinking, and facial muscle activity (Jebelli, Hwang, & Lee, 2018). After that, we can obtain channel-wise clean ECG signals.

3.2. Functional brain connectivity image construction

Different tasks activate different regions of the brain cortex. For example, performing and coordinating motor tasks activate the Sensorimotor region, complex problem-solving tasks activate the Frontal–parietal region, and visual information processing tasks activate the Occipital region (Pfurtscheller & Neuper, 1997; Yang, Deng, Xing, Xia, & Li, 2015). Recent studies find that the activated brain region shows increased local functional connectivity density, while weakly activated/deactivated regions show decreased local functional connectivity density (Tomasi, Wang, Wang, & Volkow, 2014).

Therefore, it is critical to construct a functional brain connectivity graph with vertices corresponding to neurons and edges corresponding to the temporal dependency of neurons for studying task-related brain activation. Specifically, we calculate the Pearson correlation coefficients between each pair of ΔHBO signals from n channels to get a $n * n$ adjacency correlation matrix. For each element in the matrix, high correlation coefficients represent strong functional connectivity between two neurons (i.e., corresponding channels), and low correlation coefficients represent weak functional connectivity between two neurons.

The next step is to convert the correlation matrix to an image that preserves the characteristics of functional brain connectivity. We first shift each original element value (in the range of $-1\sim1$) to $0\sim2$, and then scale it to $0\sim255$. Since blue and orange are shown to be more sensitive in deeper layers in convolutional neural networks (used in next subsection) (Flachot & Gegenfurtner, 2021), we choose jet colormap (Thyng, Greene, Hetland, Zimmerle, & DiMarco, 2016) as the mapping function from element value to RGB space. In this way, (positive and negative) strong connectivity which is represented by near-blue and near-orange can be well-captured. After that, we resize the $n * n$ to $224 * 224$ using bicubic interpolation.

3.3. Deep compression learning network

To protect users' privacy and reduce communication costs, we develop a semantic structure-based unsupervised deep compression learning network to transform functional brain connectivity images to fixed-size neural hashes. Our proposed hash function has the following characteristics:

- Irreversible: Hash values cannot be used to infer or reconstruct the original functional brain connectivity images which contain users' privacy information, such as age, gender, health condition, etc (Salakhutdinov & Hinton, 2009).
- Semantic-preserving: Relative distance between the input data samples is preserved in the relative distance between the output hash values. Therefore, similar input data samples are mapped to the same or similar hash values.

The main goal of a deep compression learning network is to map original data samples to Hamming space where semantic relationships between different data samples can be captured and preserved. Recent studies show that features extracted from pre-trained deep architectures have rich semantic information (Girshick, Donahue, Darrell, & Malik, 2014), which provides the potential to learn the semantic similarity structure without using any label information.

Semantic similarity construction: We first adopt a pre-trained VGG-F network (Simonyan & Zisserman, 2014) to extract deep feature vectors from input functional brain connectivity images. Then, we calculate the cosine distance for every two deep feature vectors. It is assumed that feature vectors with much smaller distances than the others are semantically similar, and feature vectors with much larger distances than the others are semantically dissimilar. Thereby, we split the distance histogram from the maximum value and use Gaussian distribution to approximate each part. The distance thresholds d_{similar} and $d_{\text{dissimilar}}$ for semantically similarity and dissimilarity are calculated based on the mean and standard deviation of the Gaussian distribution in each part, respectively (Yang, Deng, Liu, Liu, & Tao, 2018). After that, we can obtain a semantic similarity function as:

$$G_{jk} = \begin{cases} 1 & \text{if } d(j,k) < d_{\text{similar}} \\ 0 & \text{if } d_{\text{similar}} \le d(j,k) \le d_{\text{dissimilar}} \\ -1 & \text{if } d(j,k) > d_{\text{dissimilar}} \end{cases} \tag{1}$$

where $d(j,k)$ is the cosine distance between deep feature vectors of a image pair x_j and x_k. If an image pair is semantically similar, the semantic function will return 1; If an image pair is semantically dissimilar, the semantic function will return -1. We will set the semantic function as 0 if ambiguous similarity is obtained.

Neural hash similarity construction: In the same while, we construct a Hash similarity function by using the inner product of a pair of hash codes, which can be formulated as: $H_{jk} = \frac{1}{l}\mathbf{b}_j^\top \mathbf{b}_k$, where l is the bit number of hash codes, $\mathbf{b}_j$ is the hash codes of input image $\mathbf{x}_j$, and $\mathbf{b}_j \in \{-1,1\}^l$. If a pair of hash codes is similar, the Hash similarity function will return a value closer to 1; If a pair of hash codes is dissimilar, the Hash similarity function will return a value closer to -1.

Optimization problem: Our goal is to map semantically similar image pair into similar neural hashes, semantically dissimilar image pair into dissimilar neural hashes, and discard the pair with ambiguous semantic similarity. Therefore, we develop a loss function to minimize the difference between hash similarity and semantic similarity (Yang et al., 2018), which is formulated as:

$$min\ \Gamma(\omega) = \frac{1}{n^2}\sum_{j=1}^{n}\sum_{k=1}^{n}|G_{jk}| \cdot \|G_{jk} - \frac{1}{l}\widetilde{\mathbf{b}}_j^\top \widetilde{\mathbf{b}}_k\|_2^2, \tag{2}$$

where $\widetilde{\mathbf{b}}_j = \tanh(F(\mathbf{x}_j;\omega))$ that relaxes the binary constraint to enable the back propagation in the modified VGG-F network (i.e., the last layer is changed to a fully-connected layer with l hidden units), $F(\mathbf{x}_j;\omega)$ is l dimension output of this deep neural network, ω is the network parameters. We apply stochastic gradient descent (SGD) method (Li, Zhang, Chen, & Smola, 2014) to minimize this function to obtain optimized network parameters. Finally, for an input functional brain connectivity image $\mathbf{x}_a$, we can generate the corresponding neural hash as: $sign(F(\mathbf{x}_a;\omega))$.

3.4. Abnormal neural hash filter

After generating a neural hash for each input image, users upload the hash values and labels to the cloud. On the user end, users may accidentally generate or upload some abnormal hash values, due to the wrong measurement of brain signals or mislabeling. Therefore, we leverage a statistical autonomous and data-driven method to filter the abnormal hash values.

The first step is to identify the potential anomalies from the empirically observed hash codes by using the density-based outlier detection method. The uploaded hash codes are represented as: $\{\mathbf{b}_1,\mathbf{b}_2,\dots,\mathbf{b}_n\}$ that correspond to the input image $\{\mathbf{x}_1,\mathbf{x}_2,\dots,\mathbf{x}_n\}$. In these hash code sets, some hash codes should be the same which represent semantically similar images. The set of unique hash codes is denoted as: $\{\mathbf{u}_1,\mathbf{u}_2,\dots,\mathbf{u}_m\}$ and the corresponding frequency is defined as: $\{\mathbf{f}_1,\mathbf{f}_2,\dots,\mathbf{f}_m\}$. For a single hash code $\mathbf{b}_j = \mathbf{u}_i$, the multimodal density (Gu & Angelov, 2017) is formulated as:

$$D^M(\mathbf{b}_j) = D^M(\mathbf{u}_i) = \frac{f_i}{1+\frac{\|\mathbf{u}_i-\mu\|^2}{B-\|\mu\|}}, \tag{3}$$

where B and μ are the average dot product and mean value of the $\{\mathbf{b}_1,\mathbf{b}_2,\dots,\mathbf{b}_n\}$. Moreover, we also calculate the weighted local unimodal density (Gu & Angelov, 2017), given by:

$$D^{WL}(\mathbf{b}_j) = D^{WL}(\mathbf{u}_i) = \frac{p_i-1}{m}\cdot\frac{f_i}{1+\frac{\|\mathbf{u}_i-\lambda_i^m\|^2}{U_i^m-\|\lambda_i^m\|}}, \tag{4}$$

where m is the number of unique hash codes, p_i is the size of the $\{\mathbf{u}\}_i^m$ set, U_i^m and λ_i^m are the average dot product and mean value of the $\{\mathbf{u}\}_i^m$ set. After that, we rank $D^M(\mathbf{b})$ and $D^{WL}(\mathbf{b})$ in ascending order, respectively, and consider the first half of $\frac{1}{9}$ of the hash codes with the smallest multimodal density and weighted local unimodal density as potential anomalies.

The second step is to use the union-find method to find several clusters of these potential anomalies. Finally, we calculate the average cluster members count and compare each cluster. If the member count of one cluster is lower than the average, then the potential anomalies will be regarded as actual anomalies and be discarded in the cloud.

3.5. Customized neural decoding models

After obtaining filtered neural hashes and corresponding labels, we develop a set of decoding models that are customized for different applications. For each functional brain connectivity image, it is mapped to a single neural hash, but it can correspond to multiple types of labels. For example, a functional brain connectivity image can have a behavior label and an emotion label at the same time. Different from application-based features, neural hashes are fair to different applications and thereby only need to be uploaded once to the cloud for training diverse decoding models, which largely reduce communication costs in distributed learning. By leveraging powerful computational resources in the cloud server, we are able to select the most suitable decoding models including classic machine learning models (e.g., SVM, Random Forest, etc.) and deep learning models (e.g., CNN, LSTM, etc.) for different applications.

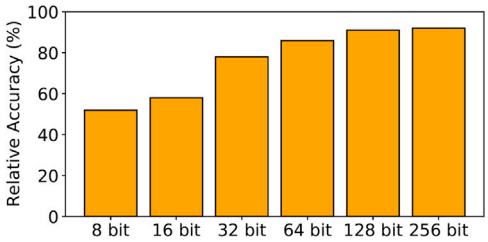

Fig. 2. The impact of hash codes length on mental arithmetic recognition performance using fNIRS signals.

4. Evaluation

4.1. Data preparation

We prepare three types of brain data (e.g., fNIRS, EEG, and fMRI) for three different applications, which are mental arithmetic recognition, motor imagery recognition, and visual stimuli recognition. With a pre-trained VGG-F network, we train a customized deep hashing model for each subject. After that, the generated hash codes and corresponding labels are used to train a recognition model in the cloud. We perform 5-fold cross-validation for each recognition model.

In the fNIRS dataset (Shin et al., 2016), fNIRS signals are collected by 72 channels at a 10 Hz sampling rate. There are 29 subjects (14 males and 15 females) in this study. Each subject performs 30 mental arithmetic tasks and 30 rest tasks with a length of 10 s for each. The mental arithmetic task is to repeatedly perform subtracting a one-digit number from a three-digit number, such as "341-8-8-..". Each task is regarded as a sample. Finally, we can get 870 mental arithmetic samples and 870 rest samples in total for training a mental arithmetic recognition model.

In EEG dataset (Shin et al., 2016), EEG signals are collected by 30 channels at a 200 Hz sampling rate. There are 29 subjects (14 males and 15 females) in this study. For motor imagery application, subjects are requested to perform haptic motor imagery, i.e. imagine the feeling of opening and closing their left hands or right hands as they were grabbing a ball. Each subject performs 30 left-hand motor imagery tasks and 30 right-hand motor imagery tasks with a length of 10 s for each. Each task is regarded as a sample. Finally, we can get 870 left-hand motor imagery samples and 870 right-hand motor imagery samples in total for training a motor imagery recognition model.

In fMRI dataset (Chang et al., 2019), the subjects are asked to judge whether they like, dislike, or are neutral about the stimuli image of each trial. One image is presented for 1 s in each trial, with 9 s of fixation between trials. There are 5252 stimuli trials in total. In each trial, the fMRI sensor scans the whole head volume five times. Since the occipital place area of the right head is activated more during the stimuli trial and is often involved in scene perception (Dilks, Julian, Paunov, & Kanwisher, 2013), we select this area that consists of 187 voxels to construct the functional connectivity map. One subject data is used in our experiment, and one trial is regarded as one sample, so we finally get 2178 samples labeled by like, 2054 samples labeled by dislike, and 1020 samples labeled by neutral for training a visual stimuli recognition model.

4.2. Metrics

We use the relative accuracy as the metric, given by:

$$Acc_R = \frac{Acc_H}{Acc_B}, \tag{5}$$

where Acc_H represents the recognition accuracy of our system, Acc_B is the baseline recognition accuracy. The baseline model is trained without using the hash codes. Specifically, each element in the functional brain connectivity matrix is regarded as a feature, and thereby we can get a n^2 dimension feature vector for inputting to an SVM classifier. For fNIRS signals and EEG signals, n is the number of channels, and for fMRI signals, n is the number of voxels. We perform 5-fold cross-validation for the mental arithmetic recognition model, motor imagery recognition, and visual stimuli recognition, and get a baseline accuracy for each.

4.3. Overall performance

We first train the deep hashing network and generate the hash code with the length of 8 bits, 16 bits, 32 bits, 64 bits, 128 bits, 256 bits, respectively. Then, we train the recognition model with each length of hash code as input. The impact of the hash codes length on system performance is evaluated.

Mental arithmetic recognition: As shown in Fig. 2, we find that the relative accuracy is low when the length of hash codes is less than 16 bits. The performance is improved when the length of hash codes increased, however, the improvement is not significant as the length of hash codes increases from 128 bits to 256 bits. To be specific, the relative accuracy can achieve 78%, 86%, 91%, 92% for 32 bits, 64 bits, 128 bits, and 256 bits, respectively.

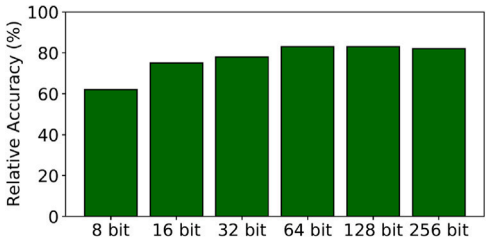

Fig. 3. The impact of hash codes length on motor imagery recognition performance using EEG signals.

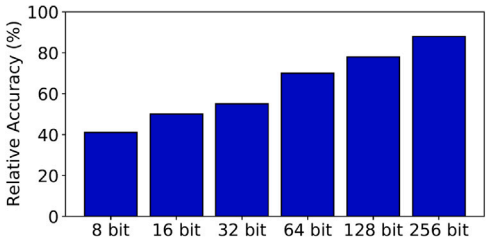

Fig. 4. The impact of hash codes length on visual stimuli recognition performance using fMRI signals.

Motor imagery recognition: The system achieves around 83% when the length of hash codes reaches 64 bits, but it does not increase a lot after that, as shown in Fig. 3. This is because the original size of the functional brain connectivity graph (i.e., 30∗30) is relatively small, and thereby does not need much hash space to store the connectivity information. The highest accuracy is much lower than the mental arithmetic recognition model trained with fNIRS data. This may be because the functional connectivity of EEG data that is based on the time-series correlation misses some frequency information in representation. Such frequency information is proved to be effective in many EEG-based neural signal analytics (Ismail & Karwowski, 2020).

Visual stimuli recognition: As observed in Fig. 4, the relative accuracy slightly increases from 8 bits to 64 bits, and then experiences a notable improvement from 64 bits to 256 bits, and reaches the highest at 256 bits, i.e., 88%. This is because the functional brain connectivity graph is 187∗187, some rich connectivity information cannot be well-preserved in a small hash size.

To summarize, the trained decoding models for different applications can achieve above 83% relative accuracy when the neural hash size is set as 128 bits. Compared with the transmission of a functional brain connectivity matrix or image, the transmission cost can be reduced by at least 100 times, which largely saves the communication cost.

4.4. Performance with demographic factors

We are curious about whether the generated neural hashes can be used to train inclusive decoding models. Therefore, we evaluate the recognition performance under different demographic parameters using the fNIRS dataset and EEG dataset, and the neural hash size is set as 128 bits. 14 males and 15 females are enrolled in the evaluation. Fig. 5 (a) shows that the relative accuracy is almost the same for male subjects and female subjects in both mental arithmetic recognition and motor imagery recognition. It indicates that the decoding models trained on neural hashes are insensitive to gender factors. In the age group, 21 subjects aging from 20–30, and 8 subjects aging above 30. As observed in Fig. 5 (b), the 20–30 and >30 age group achieve around 91.5% and 90% relative accuracy of mental arithmetic recognition, and achieve around 82.7% and 83.6% relative accuracy for motor imagery recognition. It suggests that the age factor does not affect the performance of decoding models trained on neural hashes.

5. Conclusion

In this paper, we develop deep compression learning techniques in privacy computing infrastructure for analyzing and training neural decoding models. In the edge end, the heterogeneous neural signals are first processed and converted to functional brain connectivity images with unified size. Then, we train an unsupervised deep compression learning network based on the semantics structure of the input images to generate neural hashes, which are uploaded to the cloud together with their labels afterward. Neural hashes are irreversible that cannot be used to reconstruct privacy. Moreover, the size of neural hash is small which largely reduces communication cost. In the cloud end, a statistical autonomous and data-driven approach is leveraged to filter abnormal hash codes. After that, we use these neural hashes and labels to train different decoding models. Experiment results show that the trained decoding models for different applications can achieve more than 83% relative accuracy.

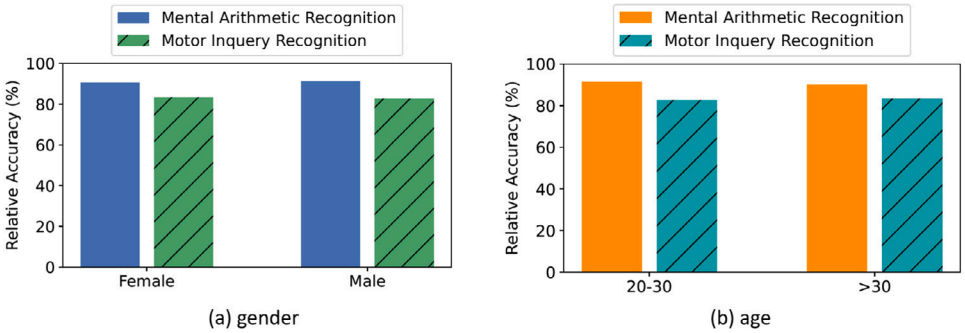

Fig. 5. The performance of trained recognition models under different demographics.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported by the U.S. National Science Foundation under Grants CNS-2050910, ECCS-2028872 and the Key Research and Development Program of Shaanxi, China (Grant No. 2021GY-054).

References

Athanasiou, V. (2014). *Estimation and modelling of fmri bold response* (Ph.D. thesis), University of Linköping.

Bercea, C. I., Wiestler, B., Rueckert, D., & Albarqouni, S. (2021). FedDis: DIsentangled federated learning for unsupervised brain pathology segmentation. arXiv preprint arXiv:2103.03705.

Chang, N., Pyles, J. A., Marcus, A., Gupta, A., Tarr, M. J., & Aminoff, E. M. (2019). Bold5000, a public fMRI dataset while viewing 5000 visual images. *Scientific Data*, *6*(1), 1–18.

Dilks, D. D., Julian, J. B., Paunov, A. M., & Kanwisher, N. (2013). The occipital place area is causally and selectively involved in scene perception. *Journal of Neuroscience*, *33*(4), 1331–1336.

Flachot, A., & Gegenfurtner, K. R. (2021). Color for object recognition: Hue and chroma sensitivity in the deep features of convolutional neural networks. *Vision Research*, *182*, 89–100.

Gao, D., Ju, C., Wei, X., Liu, Y., Chen, T., & Yang, Q. (2019). Hhhfl: Hierarchical heterogeneous horizontal federated learning for electroencephalography. arXiv preprint arXiv:1909.05784.

Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the ieee conference on computer vision and pattern recognition. (pp. 580–587).

Gu, X., & Angelov, P. (2017). Autonomous anomaly detection. In *2017 Evolving and adaptive intelligent systems* (pp. 1–8). IEEE.

Hoshi, Y. (2003). Functional near-infrared optical imaging: Utility and limitations in human brain mapping. *Psychophysiology*, *40*(4), 511–520.

Ismail, L. E., & Karwowski, W. (2020). A graph theory-based modeling of functional brain connectivity based on eeg: A systematic review in the context of neuroergonomics. *IEEE Access*, *8*, 155103–155135.

Jebelli, H., Hwang, S., & Lee, S. (2018). EEG signal-processing framework to obtain high-quality brain waves from an off-the-shelf wearable EEG device. *Journal of Computing in Civil Engineering*, *32*(1), Article 04017070.

Li, X., Gu, Y., Dvornek, N., Staib, L. H., Ventola, P., & Duncan, J. S. (2020). Multi-site fMRI analysis using privacy-preserving federated learning and domain adaptation: ABIDE results. *Medical Image Analysis*, *65*, Article 101765.

Li, M., Zhang, T., Chen, Y., & Smola, A. J. (2014). Efficient mini-batch training for stochastic optimization. In Proceedings of the 20th acm sigkdd international conference on knowledge discovery and data mining. (pp. 661–670).

McMahan, B., Moore, E., Ramage, D., Hampson, S., & y Arcas, B. A. (2017). Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics* (pp. 1273–1282). PMLR.

Melis, L., Song, C., De Cristofaro, E., & Shmatikov, V. (2019). Exploiting unintended feature leakage in collaborative learning. In *2019 IEEE symposium on security and privacy* (pp. 691–706). IEEE.

Pfurtscheller, G., & Neuper, C. (1997). Motor imagery activates primary sensorimotor area in humans. *Neuroscience Letters*, *239*(2–3), 65–68.

Salakhutdinov, R., & Hinton, G. (2009). Semantic hashing. *International Journal of Approximate Reasoning*, *50*(7), 969–978.

Santhanam, G., Ryu, S. I., Byron, M. Y., Afshar, A., & Shenoy, K. V. (2006). A high-performance brain–computer interface. *Nature*, *442*(7099), 195–198.

Shin, J., von Lühmann, A., Blankertz, B., Kim, D.-W., Jeong, J., Hwang, H.-J., et al. (2016). Open access dataset for eeg+ NIRS single-trial classification. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, *25*(10), 1735–1745.

Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.

Thyng, K. M., Greene, C. A., Hetland, R. D., Zimmerle, H. M., & DiMarco, S. F. (2016). True colors of oceanography: Guidelines for effective and accurate colormap selection. *Oceanography*, *29*(3), 9–13.

Tomasi, D., Wang, R., Wang, G.-J., & Volkow, N. D. (2014). Functional connectivity and brain activation: a synergistic approach. *Cerebral Cortex*, *24*(10), 2619–2629.

Wolpaw, J. R., Birbaumer, N., Heetderks, W. J., McFarland, D. J., Peckham, P. H., Schalk, G., et al. (2000). Brain-computer interface technology: a review of the first international meeting. *IEEE Transactions on Rehabilitation Engineering*, *8*(2), 164–173.

Wu, Q., Chen, X., Zhou, Z., & Zhang, J. (2020). FedHome: CLoud-edge based personalized federated learning for in-home health monitoring. *IEEE Transactions on Mobile Computing*.

Yan, C.-G., Wang, X.-D., Zuo, X.-N., & Zang, Y.-F. (2016). DPABI: data processing & analysis for (resting-state) brain imaging. *Neuroinformatics*, *14*(3), 339–351.
Yang, E., Deng, C., Liu, T., Liu, W., & Tao, D. (2018). Semantic structure-based unsupervised deep hashing. In Proceedings of the 27th international joint conference on artificial intelligence. (pp. 1064–1070).
Yang, Y.-l., Deng, H.-x., Xing, G.-y., Xia, X.-l., & Li, H.-f. (2015). Brain functional network connectivity based on a visual task: visual information processing-related brain regions are significantly activated in the task state. *Neural Regeneration Research*, *10*(2), 298.