# EEG EMOTION RECOGNITION BASED ON CONTRASTIVE SELF-SUPERVISED LEARNING

*Yingdong Wang*[⋆] *Qingfeng Wu*[⋆⋆] *Yilong Yang*[†]

[⋆]Informatics School of Xiamen University, Siming, Xiamen, China
[†]University of Southampton, Highfield, SO17 1BJ, United Kingdom

## ABSTRACT

Domain adaption is crucial for EEG emotion recognition, which transfers the knowledge learned from a labeled source domain to an unlabeled target domain with a different data distribution. However, the training data in the source domain is required by most traditional methods, which is usually not available for EEG emotion recognition due to privacy-preserving policies. Thus, the source-free unsupervised domain adaption (SFUDA) problem emerges. To solve this problem, this paper proposes a Contrastive Self-Supervised Learning (CSSL) framework for online calibration of EEG recognition without using source data. The CSSL framework consists of two steps. In the first step, the source shared and personalized generator and classifier are trained, and contrastive learning tricks make the shared and personalized features closer when they are from the same data, thus making features more suitable for the cluster to generate pseudo labels in the second step. In the second step, the pseudo labels generated by the cluster or Gaussian mixture modeling (GMM) are applied to supervise the training of the target model. Experimental results indicate that our proposed method performs better than the state-of-art methods, and it achieves an accuracy of $89.2\%$ and $61.6\%$ on the SEED and DEAP datasets, respectively. The code is available at https://github.com/heibaipei/DCIM.

***Index Terms*—** contrastive learning, self-learning, EEG, emotion recognition, emotion classification

## 1. INTRODUCTION

As an important emotion detection approach, electroencephalogram (EEG) is invisible and difficult to replicate, which makes EEG more reliable than other emotion detection methods [1]. However, inter-subject variance makes EEG-based emotion recognition models not friendly for new users. Also, due to the privacy protection policy, the problem of domain adaptation without using source data needs to be solved. In this case, source-free unsupervised domain adaption (SFUDA) has drawn much attention in the fields of computer vision (CV) and natural language processing, but not in the EEG signal analysis field.

As for source-free self-supervised learning methods, Zhang et al. [2] proposed a data-augmented framework to generate high-quality and high-diversity simulated EEG samples for contrastive learning. Shen et al. [3] adopted contrastive learning to minimize the inter-subject variance by maximizing the similarity in EEG signal representations across subjects when they receive the same emotional stimuli. However, all self-supervised learning methods require pretext training, and the final task is in downstream learning.

The methods for solving the SFUDA problem only use source models and unlabeled target data, and there are state-of-art methods in the CV field. In SHOT[4], a generalized model is trained with all source data as a source model. Only the parameters of the source model are transmitted, and then the target subject data is fed into the source model to generate features, which are clustered to generate pseudo tags. Meanwhile, the parameters of the feature model are updated with the clustered pseudo tags. SHOT ++[5] exploits both information maximization and self-supervised learning for feature extraction. It divides the target data into two splits according to the confidence of predictions (labeling information) and then employs semi-supervised learning to improve the accuracy of less-confident predictions. So two challenges need to be overcome for SFUDA. **One challenge is how to establish a meaningful source model**. Emotional EEG is highly correlated with personality, living environment, and culture. This paper hypothesizes that the emotional features of EEG signals can be divided into domain-invariant features and domain-specific features. **The other challenge is how to improve the quality of the pseudo-labels generated by the clusters and optimize the training process.** Papers [4, 5] applied the IM loss [6, 7, 8] to make the target outputs individually similar and globally diverse. Lee et al. [9] proposed a Confidence score Weighting Adaptation using the JMDS (CoWA-JMDS) framework to solve the SFUDA problem.

In this paper, for the cluster feature, contrastive learning is applied to align domain-invariant features and domain-specific features. It is hoped that the two feature are close when they are obtained from the same sample. For feature cluster, three state-of-the-art methods for solving the SFUDA

*Corresponding Author. Thanks for the financial support from the Key Project of National Key R&D Project(No.2017YFC1703303)

problem, namely SHOT [4], SHOT++[5], and CoWA [9] are adopted, implemented, and modified for EEG representation learning. Meanwhile, a comprehensive study is performed by comparing these methods when different pseudos labels are generated and different argument methods are applied. It is demonstrated that these methods can be effectively used in Brain-Computer Interface (BCI) for EEG emotion recognition, and features learned by minimizing the contrastive loss are more robust when generating pseudos-labels.

## 2. METHODS AND DATASETS

Let $\{(x_i^s, y_i^s)\}_{i=1}^n$ denote the dataset consisting of $s$ pieces of source data and $\{x_i^t, y_i^t\}_{i=1}^n$ denote the target dataset, where $X_i \in R^{C\times T}$ represents the processed data, $C$ represents the number of channels, $T$ represents the data length, $y$ corresponds to the condition (i.e., class) label, and $s_i \in \{1, 2, 3, ..., k\}$ denotes the subject ID.

### 2.1. Training the shared feature extractor and the individual feature extractor

As shown in Fig1, in the first part, the shared feature extractor consists of three components: $E_s$ a sharing feature extractor $f_1 = h_1(X; \theta_{f_1})$, a semantic classifier $c_1 = G_{y1}(., \theta_{y1})$, and a domain discriminator $d_1 = G_{d1}(., \theta_{d1})$. The classifier and the adversary network model satisfy the likelihoods $P_{\theta_{y_i}}(y|h)$ and $P_{\theta_{d1}}(d|h)$, respectively. To extract the sharing feature, this paper proposes an adversarial game and applies Gradient Reversal Learning (GRL) to $G_{d1}$. GRL maximizes the differential entropy $loss_{d_1}$ so that the model cannot discriminate the domains the data come from. Also, the feature extractor $f_1$ needs to avoid information loss and satisfy the likelihood $G_{y1}(., \theta_{y1})$. The following parameters are trained to fit the objective:

$$\hat{\theta_{f1}}, \hat{\theta_{y1}}, \hat{\theta_{d1}} = \arg \min_{\theta_{f1},\theta_{y1}} \max_{\theta_{d1}} \mathcal{L}(\theta_{f1}, \theta_{y1}, \theta_{d1}) \quad (1)$$

where the loss of the model is composed of two parts:

$$\begin{aligned} Loss_{sharing} = \alpha E_{h_1} E_{y1}[-\log p_{\theta_{y_1}}(y_1|h_1)] \\ +(1-\alpha) E_{h1} E_{d1}[\log p_{\theta_{d_1}}(d_1|h_1)] \end{aligned} \quad (2)$$

where $\alpha$ is a trade-off between the two loss functions, $1 > \alpha > 0$ .

Due to each person's experience and long-term thinking style, a person's emotion EEG is unique and can be used for identification [10]. For the individual feature extractor, multi-task approach is applied, the individual feature extractor $f_2 = h_2(X; \theta_{f2})$, the semantic classifier $y_2 = G_{y2}(., \theta_{y2})$ , and the domain discriminator $d_2 = G_{d2}(., \theta_{d2})$ are optimized by minimizing the loss of the label classifier and the domain classifier to fit the likelihoods $P_{y2}(y_2|h_2)$ and $P_{d_2}(d_2|h_2)$.

$$\hat{\theta_{f2}}, \hat{\theta_{y2}}, \hat{\theta_{d2}} = \arg \min_{\theta_{f2},\theta_{y2}} \min_{\theta_{d2}} \mathcal{L}(\theta_{f2}, \theta_{y2}, \theta_{d2}) \quad (3)$$

where the loss function is:

$$\begin{aligned} Loss_{individual} = \beta E_{h_2} E_{y_2}[-\log p_{\theta_{y_2}}(y_2|h_2)] \\ +(1-\beta) E_{h2} E_{d2}[-\log p_{\theta_{d2}}(d_2|h_2)] \end{aligned} \quad (4)$$

where $\beta$ is the trade-off of the two losses, $1 > \beta > 0$.

To fully utilize the two types of features, the features from the same sample should be closer than those from different samples. Inspired by CLIP [11], $b_1 = h2(x; \theta_{b_1})$ and $b_2 = h2(x; \theta_{b_2})$, where $b_1$ and $b_2$ represent the shared feature and the individual feature, respectively. The following parameters are trained to fit the objective:

$$\hat{\theta_{f1}}, \hat{\theta_{f2}}, \hat{\theta_{b1}}, \hat{\theta_{b2}} = \arg \min_{\theta_{f1},\theta_{f2},\theta_{b1},\theta_{b2}} \mathcal{L}(\theta_{f1}, \theta_{y1}, \theta_{d1}, \theta_{d2}) \quad (5)$$

where the loss function is:

$$Loss_{ctr} = \mathcal{L}(b1 \cdot b2', y') \quad (6)$$

where $y'$ is the data index in a batch, $b2'$ is the transpose of $b2$, and the $\mathcal{L}$ is the cross-entropy loss.

### 2.2. Unsupervised training without source data

**SHOT and SHOT++** In both methods, the pseudo labels are generated by k-means clustering. For k-means, the initialized center point $c_k^0$ is calculated based on classification results $C1$ and $C2$ and the combined features. The features can be fused in two approaches: plus and stack together. After the center point in the k-means is updated, the pseudos labels $\hat{y}$ are obtained, and the shared and private encoders can be updated by the loss:

$$\ell = l_{IM} - \gamma \cdot H((C(F(x_t), \hat{y}) \quad (7)$$

where $\gamma$ stands for the weight of the loss based on the clustering result. Meanwhile, the IM loss is adopted to avoid all the data clustering to a single class. $Loss_{ent}$ makes predictions more determined in a single result, while $Loss_{div}$ makes the result more dispersed among different categories. In the SHOT++, data-augmented methods are integrated into the algorithm.

$$\begin{aligned} l_{IM} &= l_{ent} + l_{div}, \\ l_{ent} &= -E(x_t) \sum_{k=1}^{K} \delta(C(F(x_t)))log\delta(C(F(x_t))), \\ l_{div} &= D_{KL}(\bar{p_k}, 1/K \cdot M_2) - logK \end{aligned} \quad (8)$$

where $K$ is the number of label classes, $\delta(C(F(x_t)))$ is the mean output embedding of the target data, and $M_2$ is a K-dimensional vector where all of the elements are 1. $\bar{p_k} = E_{x\epsilon\chi_t}[\delta(f_t^K(x_t))]$ when $f_t(x) = C(F(x)$, and $\bar{p_k}$ is the mean output embedding of the whole target domain.

**SHOT++ and CoWA** Compared with SHOT++, the COWA obtains the pseudo labels by GMM. To make the method more robust, mixup [12] is applied to both methods.

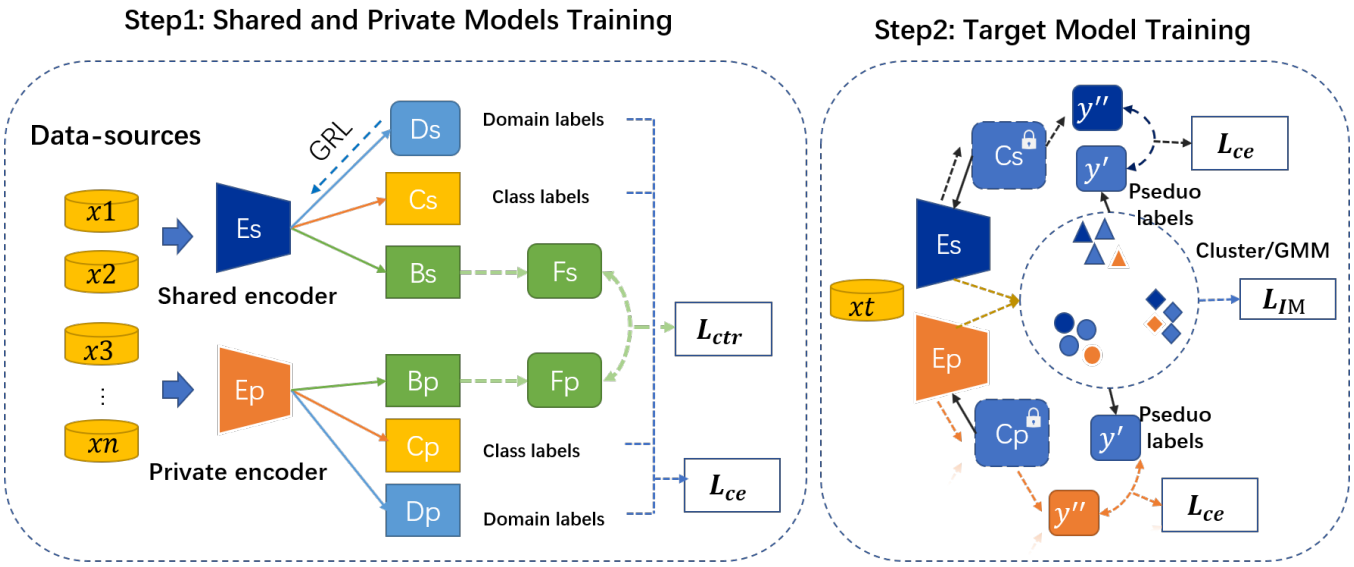


**Fig. 1**. The whole training process is divided into two stages. In the first stage, the source model is pre-trained to obtain the shared encoder $E_s$ and the private encoder $E_p$. $D_s$, $C_s$, $B_s$, $D_p$, $C_p$, and $B_p$ represent the domain classifiers, semantic classifiers, bottleneck feature extractors for source data and target data, respectively. $L_{ctr}$, $L_{ce}$, and $L_{MI}$ represent the contrastive loss (refer to Equation 6), cross-entropy loss, and IM (information maximization) loss (refer to Equation 8), respectively. The shared encoder applies gradient reversal learning to the domain loss; the private encoder applies multi-task learning and contrastive learning to two bottleneck features. In the second stage, the target sharing features and private features are combined, and then pseudo-labels are obtained through k-means or Gaussian mixture modeling (GMM); finally, supervised learning is performed with the pseudo-labels.

The unlabeled target sets are denoted as $(x^t, y^t)$, and the pseudo label is denoted as $\hat{y}$. The process is shown below:

$$\begin{aligned} x_t &= \lambda \cdot x_i^t + (1-\lambda) \cdot x_j^t \\ \hat{y^t} &= \lambda \cdot \hat{y_i^t} + (1-\lambda) \cdot \hat{y_j^t} \end{aligned} \tag{9}$$

where $\lambda \sim Beta(\alpha_1, \alpha_1)$, and $\alpha_1 \in (0, \infty)$. CoWa applies the wise weight in the training process according to the confidence score of the pseudo label, and the weight of each target sample is:

$$\omega(x^t) = \lambda \cdot JMDS(x_i^t) + (1-\lambda) \cdot JMDS(x_j^t) \tag{10}$$

and the final loss function of weight Mixup is:

$$loss_{mix} = \omega(x^t) \cdot (H((C(F(x_t), \hat{y^t}) \tag{11}$$

## 3. DATASET AND EXPERIMENT

The SEED dataset [16] contains the EEG data of 15 people (7 males and 8 females, MEAN:23.27) when they watched 15 Chinese video clips. The 15 video clips mainly induce three types of emotions, namely, neutral, positive, and negative. The duration of each video is approximately 4 minutes. The emotion EEG of each subject is recorded three times according to the scenario. Then, the public 62-channel data are down-sampled to 200 Hz and filtered from 0 to 75 Hz.

The DEAP dataset [17] is a multi-modal dataset containing the human emotional states of 32 subjects (16 males and 16 females,MEAN: 26.9). EEG signals and peripheral device signals were recorded when the subjects watched music videos. Each participant watched 40 one-minute videos and rated their emotion from five dimensions, namely arousal, valence, like/dislike, dominance, and familiarity. The lowest score is 1, and the highest score is 9. The publicly released 32-channel EEG data is downsampled to 128 Hz and filtered from 4.0 to 45.0 Hz. Each EEG trail includes a 60-second evoked EEG signal and a 3-second pre-trail baseline.

### 3.1. Implementation details

For comparison, the EEG sentiment data are divided into samples of 1 second, and all test methods adopt the leave-one-subject-out mechanism. The differential entropy (DE) reflects the degree of confusion of time series data and is considered the most important feature. In this paper, the DE features in five bands (1-4 Hz, 4-8 Hz, 8-14 Hz, 14-30 Hz, 30-50 Hz) are calculated, and the moving average filter with a 10-second window is used. In the SEED test, the size of each sample is 310 ($62 \times 5$). For this dataset, the feature extractor is a full network with three layers (310-128-64), the classifier is a one-layer network (64-3), and the domain classifier is also

**Table 1**. The results of the state-of-the-art methods

| Method | SEED | DEAP (valence) | DEAP (arousal) | DEAP (four class) |
|---|---|---|---|---|
| DGCNN [13] | 0.7995±0.09 | - | - | - |
| IAG [14] | 0.863±0.07 | - | - | - |
| DANN [15] | 0.797±0.09 | 0.4982±0.13 | 0.502±0.21 | 0.303 ±0.12 |
| Multi-task | 0.799±0.08 | 0.523±0.09 | 0.534±0.16 | 0.272 ±0.21 |
| DANN+contrastive-loss | 0.863±0.01 | 0.577±0.14 | 0.61±0.16 | 0.403 ±0.02 |
| Multi-task+contrastive-loss | 0.873±0.02 | **0.586±0.12** | 0.613 ±0.14 | **0.412 ±0.01** |
| SHOT(both) | 0.883±0.01 | 0.537±0.01 | 0.565±0.08 | 0.372 ±0.01 |
| SHOT++ | **0.892±0.02** | 0.546±0.01 | **0.616 ±0.01** | 0.383±0.01 |
| CoWA | 0.88±0.02 | 0.566±0.01 | 0.571±0.01 | 0.374±0.01 |

a one-layer network (64-14). In the DEAP dataset, the size of the input is 128 ($32 \times 4$) with no 0-4 Hz band signal. After moving the baseline, a weight decay moving average is applied to the data. For this dataset, the feature extractor is also a network with three layers (128-96-64), the classifier is a one-layer network (64-2) for two-class and (64-4) for four-class, and the domain classifier is a one-layer network (64-31). The output size of the two bottleneck feature extractors is 64. Besides, in the DEAP dataset, the threshold of the label is set to 5 for valence and arousal binary classification, and four classification labels are generated by valence and arousal labels.

The default value for k-means clustering is set to 5 cycles. Meanwhile, the parameters including alpha, beta, and gamma are set to 0.1, 0.1, and 0.3, respectively. The learning rate is set to 0.0001, and the learning rate of anti-gradient recursion for training invariant features is set to 0.1. The batch size is set to 128. Additionally, the augmented data is generated by adding noise with a standard distribution. In the experiment, the proposed model is implemented by Pytorch and runs on an NVIDIA Geforce GTX GPU.

### 3.2. Results

The compared results are presented in Table 1. The first four methods are generalized learning models. **DGCNN and IAG** both generate adaptive graphs to adapt to variance subjects. To test the importance of the contrastive loss, the classification accuracy of **DANN** and **Multi-task** learning is compared in two conditions without self-supervised learning. **With the contrastive loss and the simplest neural network, both emotion recognition methods can achieve the best accuracy on the SEED and DEAP datasets, and they even perform better than the best generalization model.** The three unsupervised adaption methods also improve the accuracy of EEG emotion recognition on the SEED dataset. SHOT++ and CoWA achieve higher accuracy than SHOT, and the mixup trick and the high entropy weight optimization could improve the robustness of the model.

On the DEAP dataset, the classification accuracy of all three SFUDA methods drops significantly, but the EEG emotion recognition accuracy of 32 individuals is more stable. It can be seen from Figure2 that if the shared classifier and the personal classifier obtain a low accuracy for emotion recognition, K-means or GMM SFUDA methods could improve the accuracy little. Since these two methods are very sensitive to the initial center point, they are difficult to be corrected if the center points are in the wrong category area at the beginning. However, self-learning, augmented style recognition, and the mixup trick could make the same type of features more compact and improve accuracy.

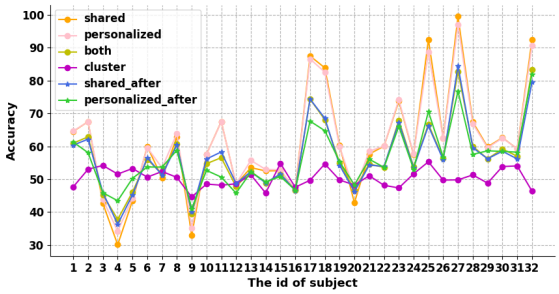


**Fig. 2**. The arousal EEG recognition accuracy of SHOT on the DEAP dataset. The "shared" and "personalized" denote the pre-training results.

## 4. CONCLUSION

This paper proposes a Contrastive Self-Supervised Learning (CSSL) framework for the online calibration of EEG recognition. Extensive experiments are conducted with three SFUDA methods, and our framework is evaluated on two public datasets. Meanwhile, the reasons for the results on different datasets are analyzed in detail. Experimental results show that the contrastive pre-trained feature model can improve the accuracy of EEG emotion classification. However, this paper only uses noise-augmented data in self-learning, and more EEG data augmentation methods will be used in future work.

## 5. REFERENCES


[1] Luca Greco, Gennaro Percannella, Pierluigi Ritrovato, Francesco Tortorella, and Mario Vento, "Trends in iot based solutions for health care: Moving ai to the edge," *Pattern Recognition Letters*, vol. 135, pp. 346–353, 2020.

[2] Zhi Zhang, Sheng-hua Zhong, and Yan Liu, "Ganser: A self-supervised data augmentation framework for eeg-based emotion recognition," *IEEE Transactions on Affective Computing*, pp. 1–1, 2022.

[3] Xinke Shen, Xianggen Liu, Xin Hu, Dan Zhang, and Sen Song, "Contrastive learning of subject-invariant eeg representations for cross-subject emotion recognition," *ArXiv*, vol. abs/2109.09559, 2022.

[4] Jian Liang, D. Hu, and Jiashi Feng, "Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation," in *ICML*, 2020.

[5] Jian Liang, Dapeng Hu, Yunbo Wang, Ran He, and Jiashi Feng, "Source data-absent unsupervised domain adaptation through hypothesis transfer and labeling transfer," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 11, pp. 8602–8617, 2022.

[6] Weihua Hu, Takeru Miyato, Seiya Tokui, Eiichi Matsumoto, and Masashi Sugiyama, "Learning discrete representations via information maximizing self-augmented training," *ICML*, 2017.

[7] Andreas Krause, Pietro Perona, and Ryan Gomes, "Discriminative clustering by regularized information maximization," *Advances in neural information processing systems*, vol. 23, pp. 775–783, 2010.

[8] Yuan Shi and Fei Sha, "Information-theoretical learning of discriminative clusters for unsupervised domain adaptation," *ICML*, 2012.

[9] Jonghyun Lee, Dahuin Jung, Junho Yim, and Sung-Hoon Yoon, "Confidence score for source-free unsupervised domain adaptation," in *ICML*, 2022.

[10] Yingdong Wang, Q. Wu, Chen Wang, and Qunsheng Ruan, "De-cnn: An improved identity recognition algorithm based on the emotional electroencephalography," *Computational and Mathematical Methods in Medicine*, vol. 2020, 2020.

[11] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever, "Learning transferable visual models from natural language supervision," in *ICML*, 2021.

[12] Hongyi Zhang, Moustapha Cissé, Yann Dauphin, and David Lopez-Paz, "mixup: Beyond empirical risk minimization," *ICLR*, vol. abs/1710.09412, 2018.

[13] Tengfei Song, Wenming Zheng, Peng Song, and Zhen Cui, "Eeg emotion recognition using dynamical graph convolutional neural networks," *IEEE Transactions on Affective Computing*, pp. 1–1, 2018.

[14] Tengfei Song, Suyuan Liu, Wenming Zheng, Yuan Zong, and Zhen Cui, "Instance-adaptive graph for eeg emotion recognition," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 3, pp. 2701–2708, 2020.

[15] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky, "Domain-adversarial training of neural networks," *The Journal of Machine Learning Research*, vol. 17, no. 1, pp. 1532–4435, 7 2016.

[16] W. Zheng and B. Lu, "Investigating critical frequency bands and channels for eeg-based emotion recognition with deep neural networks," *IEEE Transactions on Autonomous Mental Development*, vol. 7, no. 3, pp. 162–175, 2015.

[17] Sander Koelstra, C. Mühl, M. Soleymani, Jong-Seok Lee, A. Yazdani, T. Ebrahimi, T. Pun, A. Nijholt, and I. Patras, "Deap: A database for emotion analysis ;using physiological signals," *IEEE Transactions on Affective Computing*, vol. 3, pp. 18–31, 2012.