EEG-BASED BRAIN-COMPUTER INTERFACE FOR HUMAN-ROBOT COLLABORATION

BY

YAO LI

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Systems and Entrepreneurial Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2020

Urbana, Illinois

Doctoral Committee:

Professor Thenkurussi Kesavadas, Chair
Professor Henrique M. Reis
Professor Ramavarapu S. Sreenivas
Assistant Professor Girish Chowdhary

ABSTRACT

One of the expectations for the next-generation industrial robots is to work collaboratively with humans. Collaborative robots must be able to communicate with human collaborators intelligently and seamlessly. However, industrial robots in prevalence are not good at understanding human intentions and decisions. We propose to develop human-robot interactions based on Brain-Computer Interfaces (BCIs) transferring human cognition to robots directly. By collecting and encoding brain activities with BCIs, human can actively send commands to robots in thought or passively let robots monitor mental activities. We conduct two major experiments, i.e. BCI for welding robot and BCI for defective part picking robot, through which human operators can actively communicate with robots and work collaboratively on manufacturing tasks. The BCI for welding robot allows operators to select weld beads and command the robot to weld in thought. In the picking robot study, the robot picks defective part from a conveyor based on the decisions made when operators examining the qualities visually. Besides, to build faster and more accurate BCIs, we propose a Conv-CA model, which combines convolutional neural network (CNN) and canonical correlation analysis (CCA) to improve the performance of the state-of-art steady-state visually evoked potential (SSVEP) algorithm. We also conduct a study for passive BCI communication, i.e. the robot detects the circumstance when operators feel unsafe in the human-robot collaboration. When a fear response is detected, the robot can stop immediately to protect human safety.

ACKNOWLEDGMENTS

I'm extremely grateful to my advisor Professor Kesavadas for his unwavering guidance and support. More importantly, he led by example to teach me how to be a tolerant and upright person. I would like to express my deepest appreciation to Professor Reis, Professor Sreenivas, and Professor Chowdhary for their valuable advice and suggestion. I would like to extend my sincere thanks to Michelle, Tony, and Joerg for their great amount of assistance and discussion. I also had the great pleasure of working with Meiruo Xiang, Partick Wong, and Jiayi Xiang. Thanks for their helpful contributions on this project. Special thank all the experiment participants in my study. Many thanks to my friends Xiao Li, Jialin Song, Kuocheng Wang, Fanxin Wang, Donglei Sun for the encourage and help. I'm deeply indebted to my family for their unparalleled support. Especially my wife Shengzhu Shi, she is the driving force behind my research life.

This work was supported by the National Science Foundation Award Number: 1464737. I very much appreciate the support of NSF.

TABLE OF CONTENTS

CHAPTER 1 : INTRODUCTION

1.1 MOTIVATION

Robots are assisting humans in various fields, especially in industry, medical, nursing, entertainment, and our daily life. As they become more and more intelligent, robots are engaging in many new tasks which in collaboration with humans. Human-robot collaboration refers to collaborative processes in which human and robot agents work together to achieve shared goals. It can solve the automation problem that requires a large amount of human knowledge or complex action planning for the robot. For example, surgical robots perform operations collaborate with surgeons combining the medical knowledge of human and high precision motions of robots. It can also reduce the workload of programming or reprogramming while handling a variety of different objects, unpredictable materials, or working in altering environments. In manufacturing tasks like assembly, workers cooperate with robots to integrate industrial parts in specific orders with multiple tools. However, in the current human-robot collaborative interactions, human operators have to operate robots while handling the human part of the task. Ideally, we hope robots can keep a tacit understanding with operators as “robotic co-workers”. When operators make decisions, the robots should directly obtain the decisions and take appropriate actions. In this way, robots will function without extra manual operations. Besides, it would be better if robots can adapt their motion to the working status of the operator.

A direct way to obtain human's decisions is to read from the brain. Human brain activities are accompanied by bioelectrical signals. The signals can be measured from the scalp and known as electroencephalography (EEG). EEG measures voltage fluctuations resulting from ionic current within the neurons of the brain [1]. By analyzing and processing EEG, humans can control external devices with brain activities. Since 1988, brain-computer interfaces (BCIs) have been built for control robots [2]. When performing tasks, human decisions can be sent directly to the robots through BCIs. Therefore, by combing EEG-based BCIs with robots, robotic co-workers can obtain the decisions of operators through EEG when the operators are concentrating on the task without making any physical operation on the robot. While working with EEG-based robotic co-workers on industrial tasks, EEG can reveal the intentions, decisions, and mental status of workers, thereby reducing the labor and knowledge for operating the robots. Meanwhile, EEG-based robotic co-workers can provide work opportunities to disabled people who wish to work on industrial and realize their value to society.

In this study, we demonstrate that human can actively send commands to robots or passively let robots monitor mental activities through EEG-based BCIs. We use two manufacturing experiments to present the active communication. The first experiment is an EEG-controlled welding robot that can detect possible weld beads from welding parts. Operator can choose one of the detected weld beads through the BCI and initiate the robot to weld along the bead (CHAPTER 2). In the second experiment, an EEG-based robotic co-worker picks defective parts out of a conveyor according to the decisions from the human quality examiner (CHAPTER 3). Besides, in order to build more accurate and faster response BCIs, we propose a Conv-CA model, which

improves the state-of-art EEG signal classification algorithms using a convolutional neural network (CNN) and a signal processing function combined structure (CHAPTER 4). Moreover, we conduct a study on the passive human-robot BCI communication. In the study, we detect the fear response from EEG, so that, when the operator feels unsafe in the collaboration, the robot can stop immediately to protect the safety of operator (CHAPTER 5).

1.2 BACKGROUND

The challenge of robotic co-worker development comes from two aspects: physical human-robot interaction and cognitive human-robot interaction. For physical human-robot interaction, the most critical problem to be solved is how to secure human safety while collaborating with robots. Some work in hardware design (e.g. lightweight robots) [4] and safety actuation [5,6] greatly reduced the damage when the human-robot collision occurs. In terms of software, robots are programmed to avoid collision or actively make reaction in collision [3,7]. Other contributions include safety and production optimization [8,9], human safety quantification [10-12], etc. As for the cognitive human-robot interaction aspect, researches mainly focus on new human-robot interaction and human intension prediction. Gemignani et al. [13] developed a robot-operator dialog interface allowing non-expert operators to interact with the robot with voice without knowing any internal representation of the robot. Sara [14] and Brian et al. [15] studied the intuition of gestures by human observers to explore human-robot interaction gestures. However, using the voice and gesture interfaces, operators still need to operate the robot in collaboration. To eliminate human operations, Beetz et al. [16] applied artificial intelligence to analyze the

operator's intentions by predicting their motion. Olusegun et al. [17] predicted a set of human's future actions through the Kinect video stream using a CNN model.

Studies show that humans can achieve multi-dimensional robot control through BCIs with many different strategies and input modalities. Invasive BCIs can be used to implement accurate and complicated robot controls. Vogel et al. [18] demonstrated that human subjects could continuously control a robot arm to retrieve a drink container through an invasive BCI called BrainGate. However, invasive BCI requires surgery to place a chip on the brain. The EEG-based Non-invasive BCIs are capable of controlling various devices with only a short piece of EEG signal. Edlinger [19] built a virtual smart home where devices like TV, MP3 player, and phone, can be controlled through a BCI using P300 and steady-state visually evoked potentials (SSVEPs). Riaz et al. [20] and Yin et al. [21] developed a BCI language communication tool using P300 speller and speech imagery. In robot controls, Ying [22] built an on-line robot grasp planning framework using BCI to select the grasping target and grasping pose. Hortal et al. [23] trained a robot to touch one of the four target areas by detecting four different mental tasks. Gandhi et al. [24] and LaFleur et al. [25] proposed a mobile robot and quadcopter control interface for 2D and 3D navigation through motor imagery.

CHAPTER 2 : SSVEP-BASED WELDING ROBOT

2.1 OVERVIEW

We develop a welding robot application which allows the operator to select weld beads through an EEG-based BCI. The robot detects all possible weld beads from a welding part and displays each of them for operators to choose. The decisions are sent to the robot through the BCI. Then, the robot welds along the chosen bead. The robot integrates an HD camera (Logitech C920 HD Pro Webcam), a visual stimuli generator (HP Pavilion 22CWA LED Monitor), a BCI driven user interface (B-Alert X24, 20 EEG channels), an industrial manipulator (5-DoF KUKA youBot arm with a two-finger gripper), and a virtual welding simulator as shown in Figure 1. When weld on a new part, the HD camera calibrates its position and detects all possible weld beads on it. The detected weld beads are displayed on the monitor as visual stimuli to the operator. The operator can choose one of them by looking at the visual stimuli. Then, the robot welds on the chosen weld bead. In our prototype, instead of conducting actual welding, we simulate the welding process in a virtual environment.

2.2 WELD BEAD DETECTION

Weld beads are detected in three steps: 1) generate a mask for all weld beads; 2) thin the mask to a skeleton; and 3) separate each individual weld bead from the skeleton. In the first step, we apply Deeplab [26] model to generate the mask for all weld bead. Deeplab is the state-of-art

deep learning model for image sematic segmentation developed by Google. The model assigns semantic labels to every pixel in the input image. We applied the version 'DeepLabv3+' which uses an encoder-decoder structure with atrous convolution to refines the segmentation result especially along the object boundary. The model segments our input images into two classes, i.e. background and weld bead. We used Resnet50 as backbone. Because the area of weld bead is much smaller than the background, we assigned different weights for these two classes. The weight for the loss of weld bead is 0.8 and the weight for the loss of the background is 0.2. The model is trained with SGD (*lr = 7e-3, momentrum=0.9*) for 500 epochs. The training dataset contains 40 photos of 3D printed welding parts. On each part there are multiple intersecting weld beads. Figure 2 (a) demonstrates a welding part example.

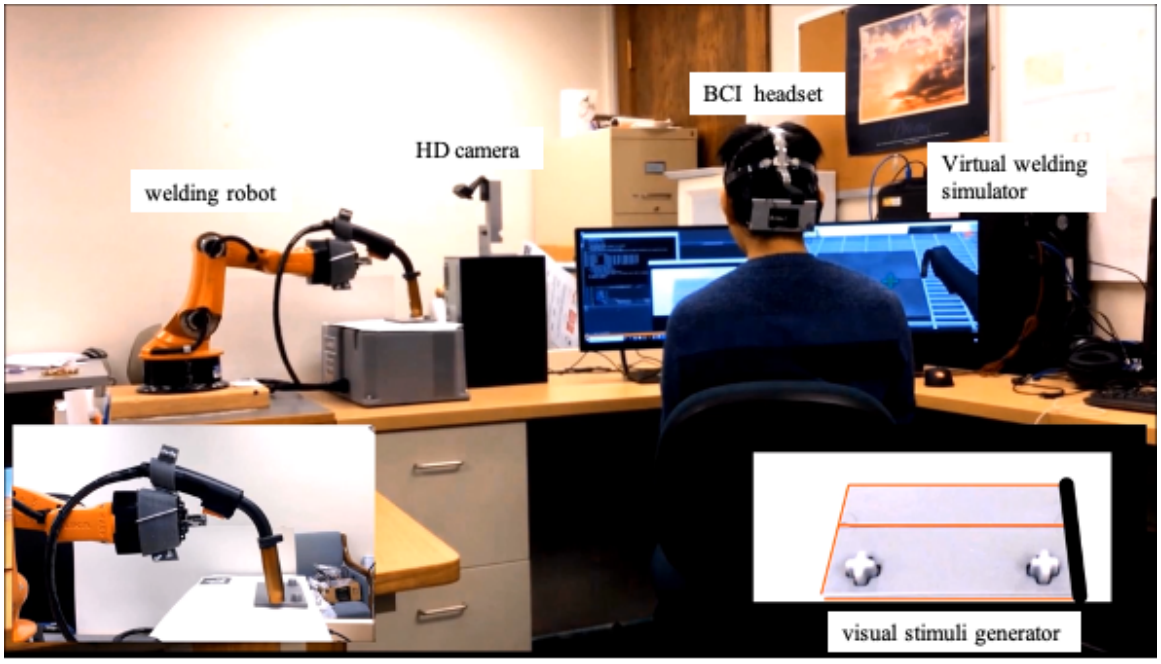


Figure 1: System overview of the welding robot application.

The mask (Figure 2 (b)) obtained from the Deeplab is further thinned to a skeleton of weld beads. The thinning approach is based on Zhang-Suen's thinning algorithm [27]. The algorithm generates a clean skeleton for the binary mask, but it contains some undesired distortions at skeleton intersection areas (necks) and corner areas (tails) (Figure 2 (c)). To eliminate distortions, we introduce a function "modify skeleton" to refine the skeleton. The “modify skeleton” function can be found in Algorithm 1. It searches the necks and tails by pairing the intersection points and end points in the skeleton and fixes them by replacing the local skeletons with spline interpolations. After the skeleton is fixed, by searching from one end point to the other end points, each individual weld bead can be separated, as described in Algorithm 2 and shown in Figure 2 (d).

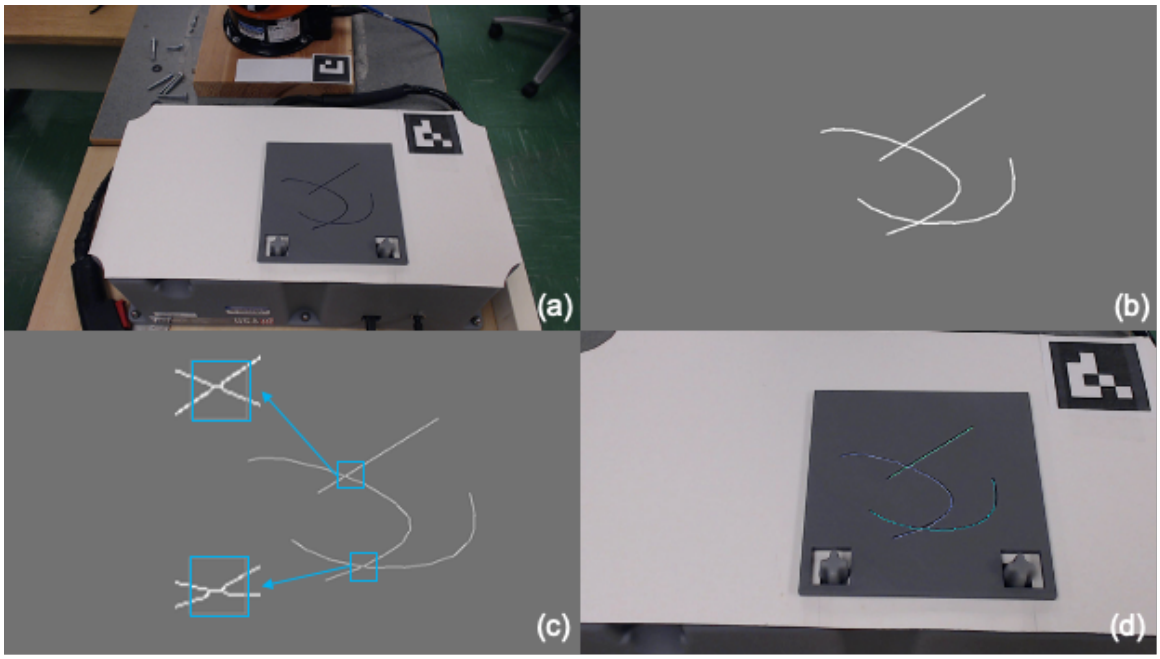


Figure 2: Weld bead detection process: (a) raw image input, (b) mask by Deeplab (zoomed in), (c) skeleton by Zhang-Suen thinning algorithm and neck distortions, and (d) separated individual weld beads.

Algorithm 1: modify skeleton

Inputs:

$\boldsymbol{I}_m$: the mask return by Deeplab

$\boldsymbol{I}_s$**:** the skeleton return by Zhang-Suen thinning

Output:

$\boldsymbol{I}_f$: fixed skeletons

$\boldsymbol{I}_f \leftarrow$ a copy of $\boldsymbol{I}_s$

$\{\boldsymbol{p}_i\} \leftarrow$ a list of intersection points in $\boldsymbol{I}_f$

$\{\boldsymbol{p}_e\} \leftarrow$ a list of end points in $\boldsymbol{I}_f$

$\{(\boldsymbol{p}_1, \boldsymbol{p}_2)\} \leftarrow$ Pair every $\boldsymbol{p}_i$ with another $\boldsymbol{p}_i$ or a $\boldsymbol{p}_e$ such that the overall travel distance of all pairs is the smallest

FOR $(\boldsymbol{p}_1, \boldsymbol{p}_2)$ in $\{(\boldsymbol{p}_1, \boldsymbol{p}_2)\}$ do

{

    IF $\boldsymbol{p_2} \in \{\boldsymbol{p_e}\}$ then ◃ fix tail areas

        Remove the skeleton between $\boldsymbol{p}_1$ and $\boldsymbol{p}_2$ from $\boldsymbol{I}_f$

    ELSE ◃ fix neck areas

    {

        $r_1, r_2 \leftarrow$ distance from $\boldsymbol{p}_1$ and $\boldsymbol{p}_2$ to the closest edge of the mask $\boldsymbol{I}_m$

        Remove the skeleton w/n radius r_1 and r_2 around $\boldsymbol{p}_1$ and $\boldsymbol{p}_2$

        Spline interpolate the remaining branches through $(\boldsymbol{p}_1 + \boldsymbol{p}_2)/2$

    }

}

Algorithm 2: separate weld beads

Inputs:

$\boldsymbol{I_f}$: fixed skeleton by modify skeleton

Output:

$\{\boldsymbol{s}\}$: a list of skeletons of separated weld beads

IF p has not been visited then

    $\boldsymbol{s} \leftarrow$: all passing points travel from p till another end point along non-adjacent branches

2.3 EEG ACQUISITION

Our BCI is a non-invasive implementation, which is an EEG-based BCI. Non-invasive BCIs yield lower performance than invasive BCIs, but they are easy to wear and demand no surgery. In our system, EEGs of operators are collected through a B-Alert X24 headset (Advanced Brain Monitoring, Carlsbad, CA), which has 20 electrodes positioned following the 10-20 system and a pair of reference channels. The sampling frequency is 256 Hz. The device is minimalistic and can be comfortably worn for an hour at a time without rewetting or reseating the electrodes.

The decisions of the operator are identified through steady state visually evoked potentials (SSVEPs). SSVEPs are natural EEG responses to visual stimulation at specific frequencies. This signal can be triggered when people looks at a flicker flashing at a stable frequency. The EEG response will have an oscillation component on the same frequency as the flicker. Based on the property of SSVEP, we display the photo of the welding part with weld beads flashing one by one

as visual stimuli. Figure 3 demonstrates a photo example of visual stimuli generation. We flash the bottom weld bead by displaying the photo (a) and photo (b) in Figure 3 alternatively. The flashing lasts 2 seconds. Then, the other four weld beads flash similarly. All the flashings are at 6 Hz. When the operator chose a weld bead, he/she only need to stare at the weld bead when it flashes. The photo is displayed on the 1920×1080-pixel LED monitor and visual stimuli are produced by Direct-X.

2.4 EEG PROCESSING

EEG is pre-processed with a 6-order band-pass Butterworth filter from 4 to 45 Hz to remove noise, muscle, eye, and movement artifacts [28]. We use canonical correlation analysis (CCA) [29] to detect the SSVEP signal. CCA is the most wildly used classification method in current SSVEP-based BCI applications. It seeks a spatial filter, which combines EEGs collected

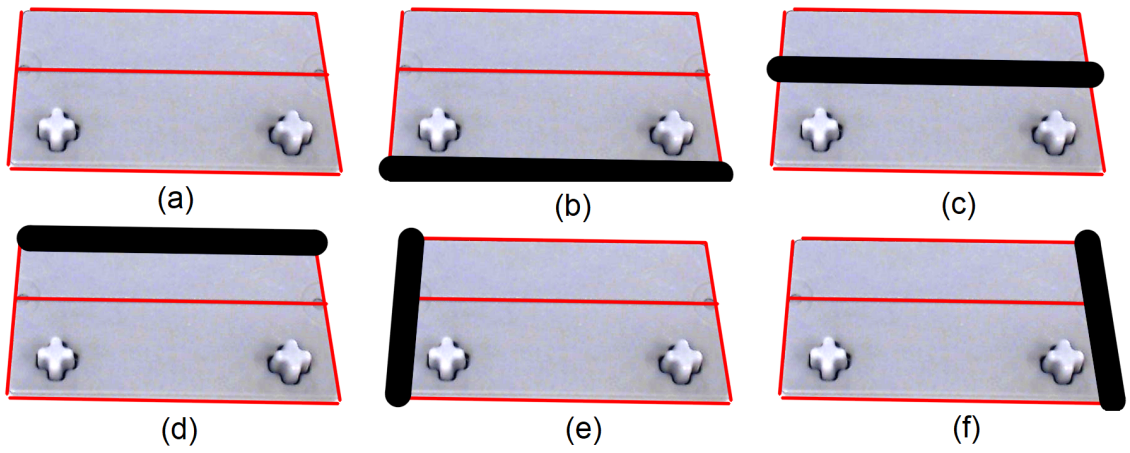


Figure 3: Example of visual stimuli generation

from multiple channels, to maximize the correlation between the combined EEG signal and a group of artificial sine and cosine signals. Assume $\boldsymbol{X} \in \mathbb{R}^{N_s \times N_c}$ is a piece of EEG data with N_s sampling points collected from N_c channels. $\boldsymbol{Y}_n \in \mathbb{R}^{N_s \times 2N_h}, (n = 1,2,\dots,N_f)$ is a group of artificial reference signals corresponding to the n-*th* stimuli frequency f_n as

$$\boldsymbol{Y}_n = \begin{bmatrix} cos(2\pi f_n \boldsymbol{t}) \\ sin(2\pi f_n \boldsymbol{t}) \\ \vdots \\ cos(2\pi N_h f_n \boldsymbol{t}) \\ sin(2\pi N_h f_n \boldsymbol{t}) \end{bmatrix}^T, \quad \boldsymbol{t} = \begin{bmatrix} \frac{1}{f_s} & \frac{2}{f_s} & \dots & \frac{N_s}{f_s} \end{bmatrix} \tag{2.1}$$

where, N_h is the number of harmonics, N_f is the number of stimuli frequencies, and f_s is the sampling frequency. CCA calculates weights $\boldsymbol{w}_x$ and $\boldsymbol{w}_y$ maximizing the canonical correlation

$$\rho = CCA(\boldsymbol{X}, \boldsymbol{Y}) = \max_{\boldsymbol{w}_x, \boldsymbol{w}_y} \frac{E\left[\boldsymbol{w}_x^T \boldsymbol{X}^T \boldsymbol{Y} \boldsymbol{w}_y\right]}{\sqrt{E[\boldsymbol{w}_x^T \boldsymbol{X}^T \boldsymbol{X} \boldsymbol{w}_x] E\left[\boldsymbol{w}_y^T \boldsymbol{Y}^T \boldsymbol{Y} \boldsymbol{w}_y\right]}} \tag{2.2}$$

To identify the frequency components of input EEG signal, CCA calculates the canonical correlations of the EEG with different artificial reference signals by $\rho_n = CCA(\boldsymbol{X}, \boldsymbol{Y}_n)$. The frequency of the reference signal which achieves the maximal canonical correlation in CCA is the classified stimuli frequency, i.e.

$$f_n^* = \underset{f_n}{argmax}\, \rho_n, \quad n = 1,2,\dots,N \tag{2.3}$$

In our experiment, all the weld beads are flashing at 6 Hz for 2 seconds. We perform CCA on each 2-seconds EEG recordings. The weld bead corresponding to the EEG recording with the highest canonical correlation is the weld bead choose by the operator.

2.5 WELD SEAM SIMULATION

We simulate the weld seam after the operator selects a weld bead. The simulation performs as a preview step for final confirmation before conducting any physical robot welding. The simulation is not based on any thermal or physical model, but only for visualization. For each pixel on the weld bead, we attach a half ellipsoid on the surface of the joint. The shape of the ellipsoid depends on the working angle θ, travel angle ϕ, welding speed v, and nozzle-plate distance d. The function of the ellipsoid is

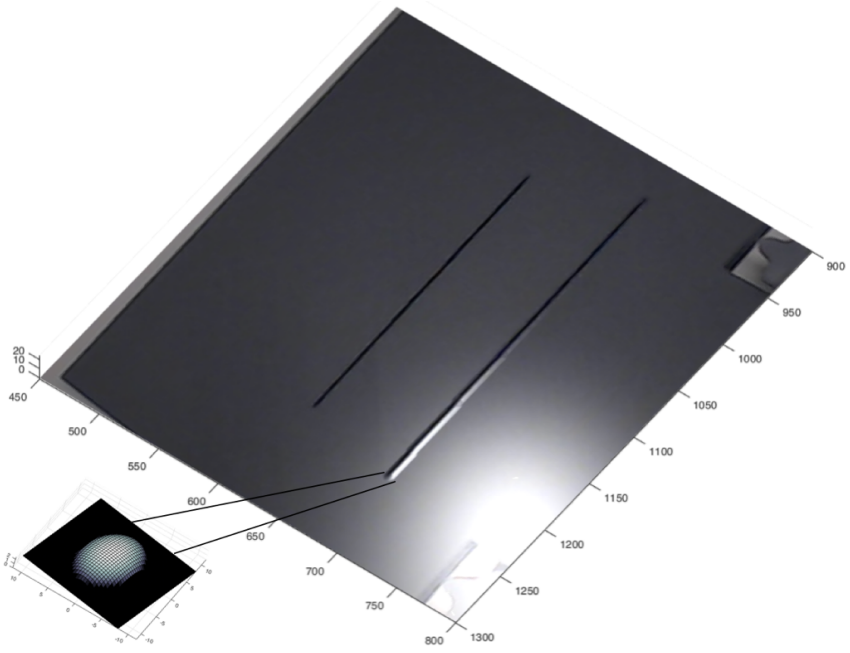


Figure 4: Welding simulation of a welding part

$$\frac{(x\cos\phi - y\sin\phi)^2}{(r/d)^2} + \frac{(x\cos\phi + y\sin\phi)^2}{(r/d\sin\theta\cos\phi)^2} + \frac{z^2}{(c/v)^2} = 1 \tag{2.4}$$

where r and c are constants. Figure 4 shows the simulation on a welding part. The section of the ellipsoid in x-y plane is determined by θ, ϕ and d. A larger nozzle-plate distance will shrink the section. Work angle and travel angle effect the length of principal semi-axes. A non-zero travel angle rotates the section as well. Besides, a faster welding speed decreases the thickness of the seam. The ellipsoids of each pixel overlap each other to form a weld. The simulation is displayed in a 3D virtual environment.

2.6 RESULTS

Three subjects participated the experiment. The subject was required to select a target weld bead from the monitor through BCI. The experiment runs 20 trials for each subject. The classification accuracy of the subject 1, subject 2, and subject 3 are 85%, 90%, and 60% by CCA method.

We use a youBot arm to demonstrate the robot welding process. Figure 5 shows the performance of robot while welding on a line in an image from the test dataset. The welding process follows butt joint weld parameters

- welding speed: 5.9267 mm/s
- travel angle: 10 degree
- work angle: 90 degree

- nozzle plate distance: 9.5250 mm

Robot end-effector trajectory, angle and speed are measured and calculated from encoders in each joint. In Figure 5, the length of the weld bead is 160 mm. The maximum offset between the robot trajectory to the extracted curve is 1.37e−3 mm. The maximum welding speed overshoot is 0.012 mm/s. For the nozzle plate distance, the maximum error is 1.52e−4 mm. The maximum travel angle error is 0.20°. The robot accurately follows the extracted welding seam and keeps welding parameters well in the demonstrated butt joint welding.

2.7 CONCLUSION AND DISCUSSION

In this experiment, we develop a BCI for welding robot which allows the operator to select a weld bead for the robot from the welding part. The weld beads are automatically detected by the robot using a deep learning model "Deeplab". The detected weld beads are displayed on the monitor and flash for 2 seconds on 6 Hz one by one. The operator needs to stare at a target weld bead for 2 seconds in order to select it for the robot. The select decision is detected from the SSVEP signal in EEG through the BCI. The EEG are segmented into 2 seconds recordings corresponding to the flashing of each weld bead. We use CCA method to calculate the canonical correlations of each EEG recording and the reference signal at 6 Hz. The weld bead corresponding to the EEG recording which has the highest canonical correlation is detected as the one chooses by the operator. Then, the robot moves along the selected weld bead and welds in the virtual environment. In our human subject study, the SSVEP classification accuracy for 3 participants are 85%, 90%,

and 60%. In the next chapter, the BCI for defective part picking robot is built based on SSVEPs in multiple frequencies. Advanced SSVEP classification algorithms are also be introduced and compared in the next chapter.

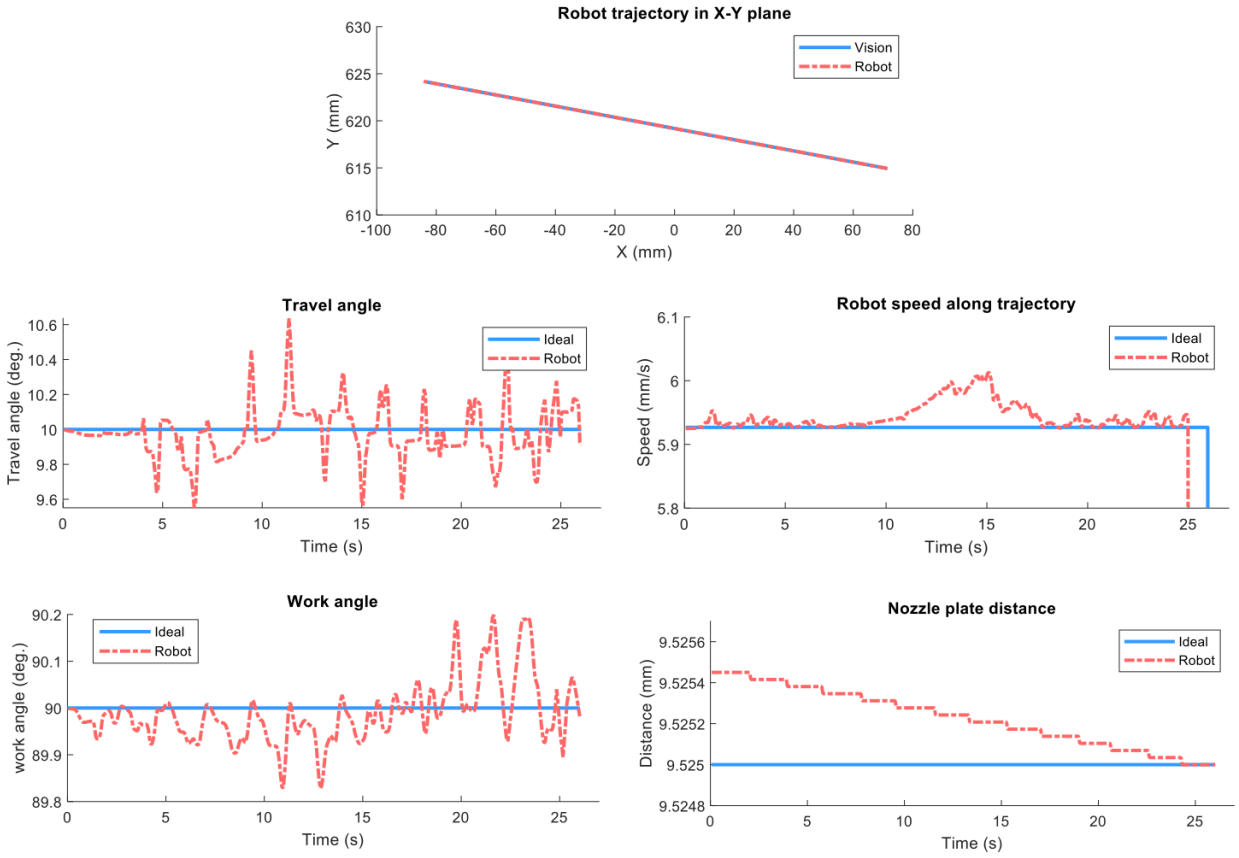


Figure 5: Demonstration of robot welding errors: robot trajectory, travel angle, work angle, speed and nozzle-plate distance.

CHAPTER 3 : SSVEP-BASED DEFECTIVE PART PICKING ROBOT

3.1 OVERVIEW

We develop a picking robotic co-worker that can remove defective parts based on the operator's decisions, which are issued directly from the operator's mind through a brain-computer interface (BCI). The robot automatically performs industrial part recognition, part position tracking, pickup gesture calculation, and trajectory planning. We demonstrated the performance of the picking robot using a human subject study. The picking robotic co-worker integrates an industrial manipulator (a 5-Dof KUKA youBot arm with a two-finger gripper), a robot vision module (two Logitech HD cameras), a DC motor conveyor, and an EEG driven BCI as shown in Figure 6. The BCI consists of an EEG collector (B-Alert X24, 20 EEG channels) and a stimuli generator (LED monitor). Denote the camera installed at the front-end of the conveyor as C_F. It observes the moment when a new industrial part loads onto the conveyor and takes a photo simultaneously. The monitor represents the photo to the operator as visual stimuli. From this photo, the operator determines the part quality in mind. Then, the decision about the qualification, which analyzed from EEG, is collected and sent to the robot. The robot arm will pick the part out of the conveyor if the decision tells the part is defective. The camera installed at the rear end of the conveyor (C_R), helps the robot arm find and pick the defective part.

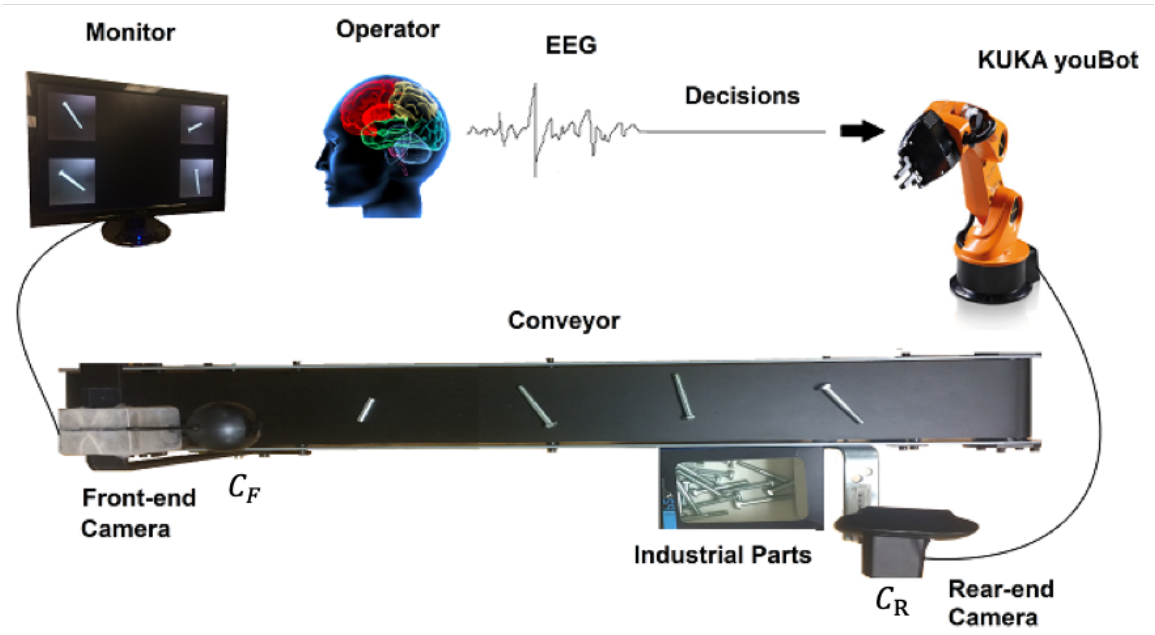


Figure 6: System diagram of the picking robotic co-worker

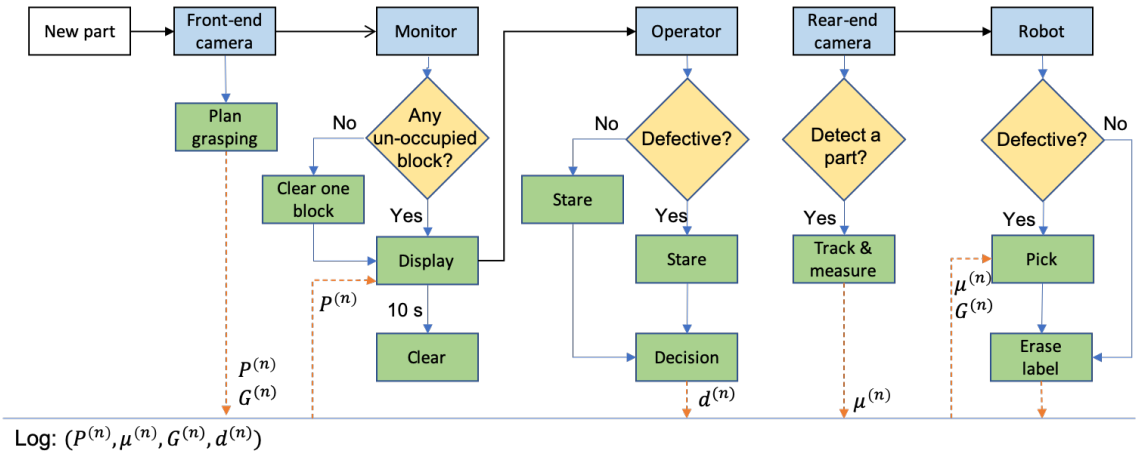


Figure 7: System workflow of the picking robotic co-worker

The detailed workflow is presented in Figure 7. When C_F detects a newly loaded part, the system registers the part in the Log thread as the *n-th* part loaded on the conveyor. C_F takes a photo ($\boldsymbol{P}^{(n)}$) for the part and stores the photo in the Log thread. The monitor is programmed to have four blocks arranged as a 2×2 matrix to display photos. $\boldsymbol{P}^{(n)}$ will fill in a block to present to the operator. If all the four blocks are filled, the monitor will clear a block to make it available. $P^{(n)}$ appears for 10 seconds, and then automatically removed from the block. During the 10 s, the operator can inspect the quality of the displaying part from the monitor. Meanwhile, the operator's EEG is collected and analyzed. The EEG is classified to be a binary decision $d^{(n)}$ and stored in the Log thread as well. Here, we have $d^{(n)} = 0$ for the part is qualified or $d^{(n)} = 1$ for the part is defective. When the part moves to the end of the conveyor, and if $d^{(n)} = 1$, C_R will extract its position $\boldsymbol{\mu}^{(n)}(x, y)$ in real-time to provide closed-loop feedback for youBot to remove it. While the robot picks the part, it follows the grasping plan $G^{(n)}$ calculated when the part passes C_F.

The operator sits in front of the monitor to inspect the qualities of industrial parts through the photos. Once he/she identifies a defective part, the operator should stare at the photo until it is marked with a green square (detection succeed) or until the photo vanishes (detection failed). If the part is qualified, the operator should avoid staring at the photo for more than 2 seconds. Otherwise, it may result in a false-positive for defective identification. In other words, the system will mark the part that the operator has stared at for more than 2 seconds as defective.

3.2 PART DETECTION AND EXTRACTION

In the photo acquired by C_F or C_R, industrial parts are extracted from background using a thresholding method. In pre-processing, a morphologically open operation with a disk (10 pixels radius) shaped structuring element is applied to the photo to remove possible lighting effects. Pixels with intensity lower than 0.2 are classified as background. In addition, all the tiny connected areas whose area is less than 20 pixels are removed to reduce noise. The rest area is the extracted parts denoted as $\boldsymbol{S}$. Figure 8 demonstrates an example of the part extraction process. Figure 8 (a) shows a photo obtained by C_R and Figure 8 (b) shows the extracted part $\boldsymbol{S}$.

We use a square signal w to perceive the time when a part loads onto the conveyor or passes the end of the conveyor. w is defined as the projection of S along the row axis, i.e. $\boldsymbol{w}_j = \max_i \boldsymbol{S}_{ij}$. For C_F, when new a step down (from 1 to 0) is observed in $\boldsymbol{w}$, the system registers a new part. Similarly, for C_R, if a new step down appears in $\boldsymbol{w}$, the system erases a registered part. Besides, to divide multiple parts in the same photo, we cut the photo at the midpoints from each step down to step up of $\boldsymbol{w}$. So that $\boldsymbol{S}$ is divided into several sub-regions with each sub-region contains only one part. We denote the sub-regions containing the $n\text{-}th$ part as $\boldsymbol{S}^{(n)}$. This is a fast way to track the number of parts on the conveyor, but there must be some space between adjacent parts along the moving direction of the conveyor. The signal in Figure 8 (c) is $\boldsymbol{w}$ corresponding to the photo in Figure 8 (a). The divided sub-regions are shown in Figure 8 (d).

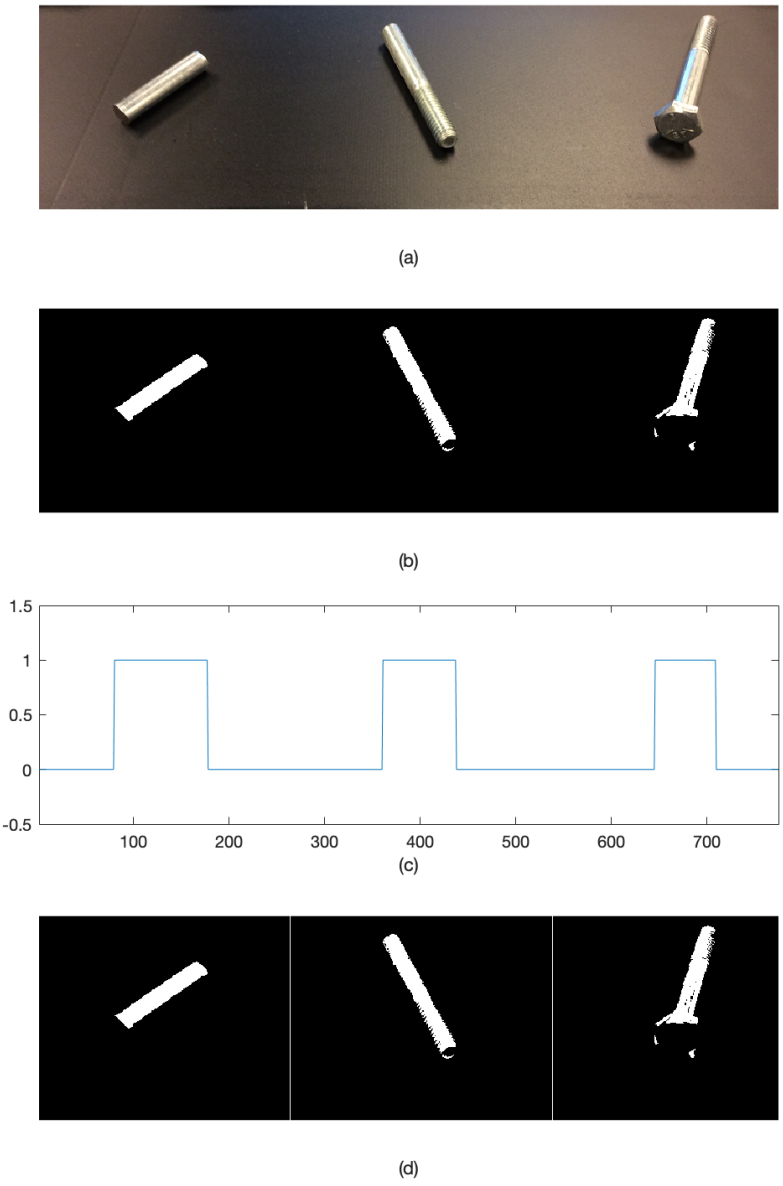


Figure 8: Part detection and extraction

3.3 PART POSITION ESTIMATION AND GRASPING PLANNING

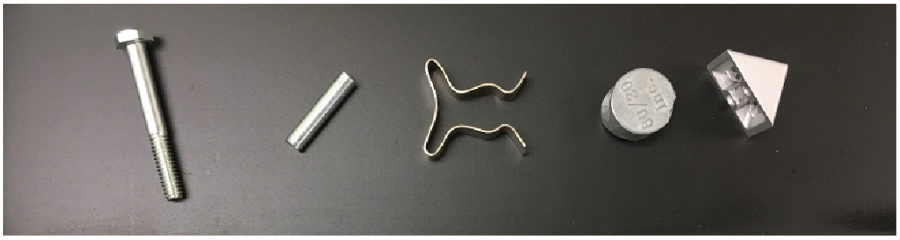
We use the geometric center ($\boldsymbol{\mu}^{(n)}$) of $\boldsymbol{S}^{(n)}$ to estimate the position of the *n-th* part. $\boldsymbol{\mu}^{(n)}$ can be easily calculated by averaging all the pixels in $\boldsymbol{S}^{(n)}$. To pick various industrial parts with the robot gripper, we developed a generalized grasping algorithm named two-finger gripper random grasp (2FRG). The 2FRG algorithm generates robust 3-Dof picking gestures for the two-finger gripper to pick unknown objects. It randomly samples N lines as N potential finger moving directions. Along each line, it searches all possible finger grasping positions (GPs). We define a GP as the position where both fingers touch the object. Then, it only keeps the GP constructing a firm grasp, which is a firm grasping position (FGP). FGP is the position that each finger has at least two points on both edges touching the object or has at least one point in the middle touching the object. We use the distance between an FGP and $\boldsymbol{\mu}^{(n)}$ as the score to evaluate the FGP. 2FRG returns the FGP with the highest score. A demonstration of a GP and an FGP are shown in Figure 9 (a) and (b). In Figure 9 (a), the right finger touches the part with only one point on the edge (red dot in the subfigure), so it is not an FGP. In Figure 9 (b), both fingers have at least one point in the middle that touches the object. Thus, the GP in Figure 9 (b) is an FGP.

We tested 2FRG on various shaped industrial parts. In Figure 10, 2FRG returns the optimal FGPs for the five different parts by sampling 200 lines. The returned FGPs are closed to geometric centers. And the grippers grasp the objects in comfortable directions. A pseudocode of 2FRG is presented in Algorithm 3. 2FRG returns the finger positions $(\boldsymbol{g}_1, \boldsymbol{g}_2)$ of the optimal FGP from N

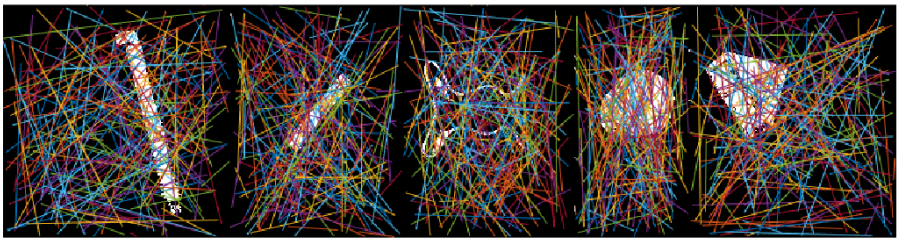
line samples. The grasping plan $G^{(n)}(\varphi, \boldsymbol{\delta})$ consists of the grasping direction φ and the offset $\boldsymbol{\delta}$ from the FGP to $\boldsymbol{\mu}^{(n)}$. Note that φ is the angle of the vector $\boldsymbol{g}_1 - \boldsymbol{g}_2$ and $\boldsymbol{\delta} = \boldsymbol{g}_1/2 + \boldsymbol{g}_2/2 - \boldsymbol{\mu}$.

(a) (b)

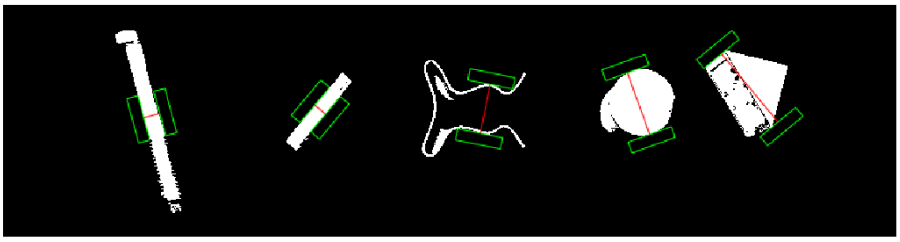
Figure 9: (a) A GP but not an FGP, and (b) an FGP.



(a)



(b)



(c)

Figure 10: Grasping planning for five different industrial parts. (a) Photo of industrial parts; (b) 200 grasping direction samples; (c) optimal FGP returned by 2FRG.

Algorithm 3: Two-finger Gripper Random Grasp (2FRG)

Inputs:

$\boldsymbol{S}^{(n)}$: region of an object stored in a segmented binary image $\boldsymbol{I}$

N: number of maximum sampling times

L: geometric constrains of a gripper

Output:

$(\boldsymbol{g}_1, \boldsymbol{g}_2)$: positions of gripper fingers

Do for N times

{

Uniformly sample two points $\boldsymbol{q}_1$ and $\boldsymbol{q}_2$ in $\boldsymbol{I}$

Construct a line segment $\boldsymbol{l} \in \boldsymbol{I}$ connecting $\boldsymbol{q}_1$ and $\boldsymbol{q}_2$ and create $\boldsymbol{g}_1, \boldsymbol{g}_2 \in \boldsymbol{l}$

$\{\boldsymbol{l}_m\} \leftarrow$ The collection of all continues sub-intervals of $\boldsymbol{l}$ such that L is satisfied if a gripper $\boldsymbol{g}_1$ or $\boldsymbol{g}_2 \in \boldsymbol{l}_m$

IF $\{\boldsymbol{l}_m\}$ is not empty then

{

$\mathrm{G_p} \leftarrow$ The collection of $(\boldsymbol{g}_1, \boldsymbol{g}_2)$ with $\boldsymbol{g}_1 \in \boldsymbol{l}_{m1}$, $\boldsymbol{g}_2 \in \boldsymbol{l}_{m2}$ and $\boldsymbol{l}_{m1}, \boldsymbol{l}_{m2} \in \{\boldsymbol{l}_m\}$ such that $(\boldsymbol{g}_1, \boldsymbol{g}_2)$ is a GP.

$\mathrm{G_{sp}} \leftarrow$ The collection of $(\boldsymbol{g}_1, \boldsymbol{g}_2) \in G_p$ such that $(\boldsymbol{g}_1, \boldsymbol{g}_2)$ is an FGP.

}

}

Return the $(\boldsymbol{g}_1, \boldsymbol{g}_2) \in G_{sp}$ with the highest score, where

$$score = -norm_{L2}\left(\boldsymbol{g}_1/2 + \boldsymbol{g}_2/2 - mean\left(\boldsymbol{S}^{(n)}\right)\right)$$

3.4 ROBOT MOTION PLANNING AND CONTROL

If the *n-th* part is defective and enters the view of C_R, the robot will pick it out based on $G^{(n)}(\varphi, \boldsymbol{\delta})$ and the $\boldsymbol{\mu}^{(n)}$ obtained by C_R in real-time. The robot moves to a pre-designed position above the conveyor with its last joint perpendicular to the conveyor surface in advance. Then the robot follows the part and moves downward to the conveyor surface. The robot velocity in the horizontal directions ($\boldsymbol{v}_{xy}$), vertical direction (v_z), and angular velocity of the last joint (α) are controlled by

$$\boldsymbol{v}_{xy} = \boldsymbol{K}_1\left(\boldsymbol{\mu}^{(n)} + \boldsymbol{\delta} - \boldsymbol{p}_{xy}\right) + \boldsymbol{K}_2\dot{\boldsymbol{\mu}}^{(n)} + \boldsymbol{K}_3\left(\ddot{\boldsymbol{\mu}}^{(n)} - \dot{\boldsymbol{v}}_{xy}\right) \quad (3.1)$$

$$v_z = k_4(z_i - p_z) - k_5\dot{v}_z \quad (3.2)$$

$$\alpha = k_6(\varphi - \theta) - k_7\dot{\alpha} \quad (3.3)$$

Here, $\boldsymbol{p}_{xy}$ and $\boldsymbol{p}_z$ are the position of robot end-effector in horizontal axes and vertical axis. θ is the last joint angle of the robot. z_i is the height of the conveyor belt. $\boldsymbol{K}_1$, $\boldsymbol{K}_2$, $\boldsymbol{K}_3$, k_4, k_5, k_6, k_7 are control gains, where $\boldsymbol{K}_1$, $\boldsymbol{K}_2$, *and* $\boldsymbol{K}_3$ are 2×2 diagonal matrixes.

3.5 EEG ACQUISITION

The BCI implementation is similar as the BCI we develop in the welding robot experiment. But instead of using only one frequency, we flash the photos of industrial parts at different frequencies. Then we can recognize which photo the operator is staring at from the EEG by distinguishing the frequencies of SSVEP. When there is no defective part presenting on the

monitor, the operator should look over all the displaying photos, or look at a blank spot. By doing this, no significant frequency components at the photos' flashing frequencies can be observed from the EEG. We call this situation as the idle state (IS). On the other hand, when the operator identified a defective part, he/she should stare at the photo until it is marked. In this case, a significant frequency component at the flashing frequency of the photo should be found in EEG.

The monitor is programmed to display visual stimuli for generating SSVEPs. The monitor displays photos in four square blocks. As shown in Figure 11, the size of the blocks is 300 pixels×300 pixels. The four blocks flash at 6 Hz, 6.67 Hz, 7.5 Hz, and 8.57 Hz, respectively. Once C_F observes a new part, the monitor will flash the photo of the part in a block. The photo of the next observed part will present in the next available block. Photos present for 10 seconds in the block. However, the monitor can display up to four photos at the same time. When C_F observes a newly loaded part and all blocks are filled, the monitor will clear the earliest filled block for the part.

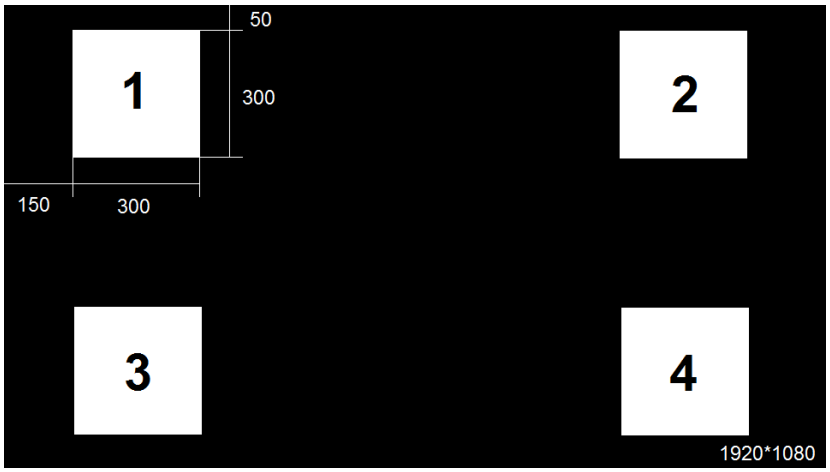


Figure 11: Blocks on the monitor

3.6 EEG PROCESSING

We compare six SSVEP classification methods, which are canonical correlation analysis (CCA), individual templated based CCA (IT-CCA) [30], support vector machine method (SVM), power spectral density-based SVM method (PSD-SVM) [31], CCA-based SVM method (CCA-SVM), and convolutional neural network CCA method (CNN-CCA). CCA-SVM and CNN-CCA are our proposed methods. EEG is pre-processed with a 6-order band-pass Butterworth filter from 4 to 45 Hz to remove noise, muscle, eye, and movement artifacts.

3.6.1 CCA method

Recall the CCA method from equation (2.3). In our application, there are four target SSVEP frequencies, which are 6 Hz, 6.67 Hz, 7.5 Hz, and 8.57 Hz, corresponding to the flashing frequencies of the blocks on the monitor. Besides, we need to classify IS as well. Therefore, EEGs will be classified as one of the five classes, i.e. SSVEP in 0 Hz (IS), 6 Hz, 6.67 Hz, 7.5 Hz, and 8.57 Hz. However, the standard CCA method can only identify non-zero frequencies. To classify the IS, we extend the standard CCA by thresholding the maximal canonical correlation. Equation (2.3) becomes

$$f_n^* = \begin{cases} \underset{f_n}{argmax}\, \rho_n, & if\ max\, \rho_n > \delta_n \\ 0, & if\ max\, \rho_n \leq \delta_n \end{cases} \qquad (3.4)$$

The threshold δ_n is searched in a training dataset such that δ_n maximize the classification accuracy of the training dataset.

3.6.2 IT-CCA method

IT-CCA is based on the CCA method but using individual template signals as reference signal instead of artificial reference signals. The individual template signal ($\overline{\boldsymbol{X}}_n \in \mathbb{R}^{N_s \times N_c}$) for the n-th stimuli is obtained by averaging the training dataset across all the trials. The canonical correlation is calculated by $\rho_n = CCA(\boldsymbol{X}, \overline{\boldsymbol{X}}_n)$. The rest of IT-CCA follows the same steps of the CCA method.

3.6.3 SVM, PSD-SVM and CCA-SVM methods

The SVM method use SVM model with 2-nd order polynomial kernel and L-2 regularization to classify EEG into the five classes. Considering SSVEP has excellent properties in the frequency domain, PSD-SVM uses the power spectra density of EEG as the feature of a standard SVM model. Similarly, CCA-SVM applies the canonical correlations of CCA as the feature of SVM. The feature is

$$\emptyset = \{\rho_n^h, \rho_\alpha\}, \quad n = 1,2,\dots,N_f, h = 1,2,\dots,N_h, \alpha = 1,2,\dots,N_\alpha \tag{3.5}$$

where

$$\rho_n^h = CCA\left(\boldsymbol{X}, \begin{bmatrix} cos(2\pi h f_n \boldsymbol{t}) \\ sin(2\pi h f_n \boldsymbol{t}) \end{bmatrix}\right) \tag{3.6}$$

$$\rho_\alpha = CCA\left(\boldsymbol{X}, \begin{bmatrix} cos(2\pi f_\alpha \boldsymbol{t}) \\ sin(2\pi f_\alpha \boldsymbol{t}) \end{bmatrix}\right) \tag{3.7}$$

Two frequencies 8 Hz and 10 Hz are chosen for f_α. The feature ρ_α measures brain activities in the alpha-band, which reflect the idle activities of the brain.

3.6.4 CNN-CCA method

CNN-CCA combines the CNN structure and CCA. The CNN structure convolutes multi-channel EEG in short sampling intervals to construct a single-channel signal. The CCA-layer at the end of the CNN framework eliminates the noise in the signal and extracts frequency features of the signal. CNN-CCA takes the EEG $\boldsymbol{X} \in \mathbb{R}^{N_s \times N_c \times 1}$ as input. It applies a 3-layer CNN convoluting $\boldsymbol{X}$ to

$$\overline{\boldsymbol{x}} = f(\boldsymbol{X}), \quad \overline{\boldsymbol{x}} \in \mathbb{R}^{N_s \times 1} \tag{3.8}$$

Here $f(.)$ is the 3-layer CNN. Then, a CCA-layer is added after the CNN-layer. The CCA-layer applies standard CCA to $\overline{\boldsymbol{x}}$ as $\rho_n = CCA(\overline{\boldsymbol{x}}, \boldsymbol{Y}_n)$, where $\boldsymbol{Y}_n$ is the reference signal corresponding to the *n-th* stimuli constructed as Equation (2.1) with $N_h = 8$, $N_f = 5$, $f_1 = 0$ (IS), $f_2 = 6$, $f_3 = 6.67$, $f_4 = 7.5$, and $f_5 = 8.57$. Note that, by Cauchy-Schwarz inequality, the maximum correlation in the CCA-layer can be calculated by

$$\rho_n = CCA(\overline{\boldsymbol{x}}, \boldsymbol{Y}_n) = \sqrt{\frac{(\overline{\boldsymbol{x}} \circ \overline{\boldsymbol{x}})^T (\boldsymbol{Y}_n \circ \boldsymbol{Y}_n)}{\overline{\boldsymbol{x}}^T \overline{\boldsymbol{x}} \boldsymbol{Y}_n{}^T \boldsymbol{Y}_n}}, \quad n \in [1, N_f] \tag{3.9}$$

where the operation $\circ$ represents Hadamard product (i.e. the element-wise product). Then, the output of the CCA-layer is

$$\boldsymbol{z} = g(\bar{\boldsymbol{x}}, \boldsymbol{Y})$$
$$= \left[CCA(\bar{\boldsymbol{x}}, \boldsymbol{Y}_1) \quad CCA(\bar{\boldsymbol{x}}, \boldsymbol{Y}_2) \quad \cdots \quad CCA\left(\bar{\boldsymbol{x}}, \boldsymbol{Y}_{N_f}\right)\right]$$
$$= [\rho_1 \quad \rho_2 \quad \cdots \quad \rho_{N_f}] \in \mathbb{R}^N \qquad (3.10)$$

Here, $g(.)$ is the CCA-layer. We use a dense layer with N_f units and softmax activation function as the final layer before classification.

Because the CCA-layer can provide non-linear operations, the CNN-layers are active by linear activation functions. The first layer of the CNNs has 16 filters of 16×4 kernels. It convolutes EEGs in all input channels ($N_c = 4$) in a short local time period (16 sampling points or 23.4 ms). The second layer combines the 16 filters in the first layer together It uses 1×4 kernels to weight EEGs from different channels. The third layer applies an 1×4 kernel with no padding (The first and second layers use zeros paddings to keep outputs the same size as the inputs.) to transform the data $\boldsymbol{X} \in \mathbb{R}^{N_s \times N_c \times 1}$ into a one-dimension signal $\bar{\boldsymbol{x}} \in \mathbb{R}^{N_s \times 1}$. At the end of the CNN-

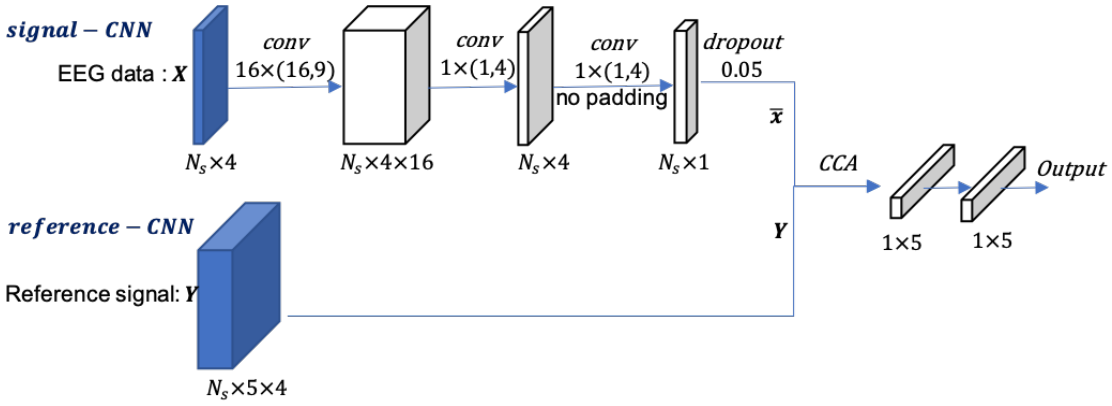


Figure 12: Structure of the CNN-CCA model

layers, we apply a dropout with dropping rate 5% to $\bar{x}$ for regularization. The detailed structure is shown in Figure 12.

The CNN-CCA is implemented in python-Keras with tensorflow backend. We use categorical cross-entropy as the loss function. The optimization is solved with Adam algorithm (learning rate (1e-4), beta1 (0.9), beta2 (0.999), gradient clipping (5)) with batch size 32.

3.7 EXPERIMENTAL SETUP

We established an offline experiment to test the performances of the above SSVEP classification methods. The experiment required subjects to stare at a flashing photo for 15 seconds in each trial. The photo flashed at one of the frequencies of 0 Hz, 6 Hz, 6.67 Hz, 7.5 Hz, and 8.57 Hz. The experiment took five runs with five trials in each run. During the experiment, subjects wearing an EEG-based BCI headset were required to stay still and blink as less as possible. Five subjects (age 25-35, 4 males, 1 female) attended the experiment.

After the offline experiment, subject 1, 2, and 3 participated in the online experiment. Subjects need to identify 2 defective industrial parts from 10 parts. The parts were manually placed on the conveyor in random order by another operator. The online experiment took three runs. All subjects successfully accomplished the task. Figure 13 shows the user interface and hardware in the online experiment.

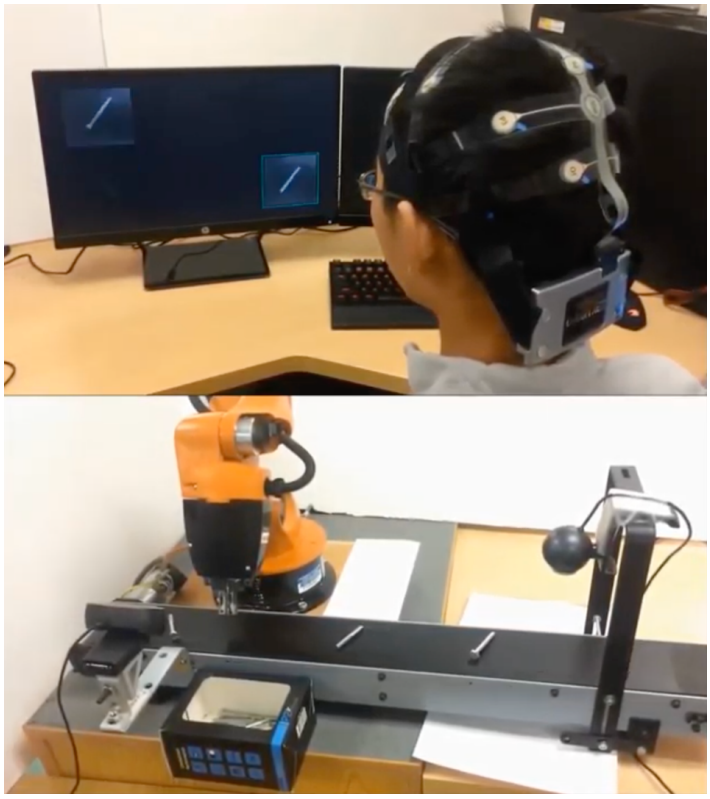


Figure 13: On-line experiment

3.8 RESULTS

Classification accuracy is used to evaluate the performance of all methods. We evaluated all the methods in four data lengths, i.e. 0.5 s, 1.0 s, 1.5 s, and 2.0 s. The data length will be mentioned as time window length (TW) in below. The data is extracted with a step of $0.15 \times \text{TW}$. We use leave-one-out cross validation to evaluate the methods. Specifically, one of the 5 trials are used for test and the other 4 trials are the training dataset. This process repeated for 5 times so that every trial is tested.

Table 1: Average classification accuracies of CNN-CCA, CCA-SVM, PSD-SVM, CCA, IT-CCA and SVM at 0.5 s, 1.0 s, 1.5 s, and 2.0 s time window lengths

TW (s)		0.5	1.0	1.5	2.0
Accuracy (%)	**CNN-CCA**	**80.24**	**89.98**	**93.63**	**96.56**
	CCA-SVM	63.98	80.14	86.67	89.73
	PSD-SVM	74.22	81.71	82.98	82.07
	CCA	48.81	66.98	77.38	83.64
	IT-CCA	53.55	63.96	68.43	70.80
	SVM	64.16	76.55	81.82	84.56

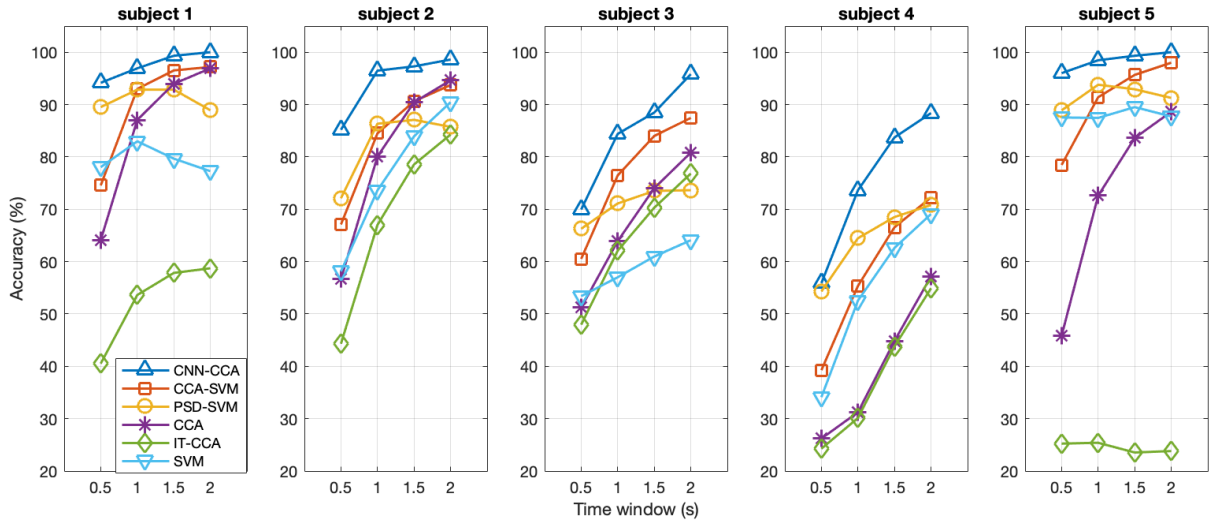


Figure 14: Classification accuracies of CNN-CCA, CCA-SVM, PSD-SVM, CCA, IT-CCA, and SVM at 0.5 s, 1.0 s, 1.5 s, and 2.0 s time window lengths for each subject in the offline experiment

As shown in Table 1, the classification accuracies of our proposed CNN-CCA method are 80.24%, 89.98%, 93.63%, and 96.56% at the 0.5 s, 1.0 s, 1.5 s, and 2.0 s TW, respectively. CNN-CCA achieved the highest classification accuracies in the comparison with CCA-SVM, PSD-SVM, CCA, IT-CCA, and SVM. Comparing to the most commonly used CCA method, CNN-CCA improved the average classification accuracies by 31.43%, 23.00%, 16.25%, and 12.92% at

0.5 s, 1.0 s, 1.5 s, and 2.0 s TW, respectively. Figure 14 shows the classification accuracies of each individual subject. CNN-CCA outperformed other tested method at 0.5 s, 1.0 s, 1.5 s, and 2.0 s TW with $p < 0.001$.

In our implementation, the most frequent class is IS (i.e. $\mathrm{n} = 1$), because the industrial parts on the conveyor are qualified in most cases. Thus, the classification accuracy of IS is more important than the other classes. To check the performance for classify IS, we calculated the confusion matrices of CNN-CCA, PSD-SVM, CCA, and CCA-SVM in Figure 15. The classification accuracies of IS are marked on the confusion matrixes. At the 0.5 and 1.0 TW, PSD-SVM has the highest IS classification accuracy. However, the classification accuracies of all the five classes are 74.22% and 81.71%. As we lengthen the TW, CNN-CCA exceed PSD-SVM to become the best IS classification model with 95% accuracy at 2.0 TW.

The result of applying CNN-CCA on the online experiment is shown in Figure 16. In the online experiment, SSVEP is classified in 2 s TW. Each five rectangles in a row represents five windows of EEG data for SSVEP classification during an industrial part flashing on the monitor. The frequency on each row of rectangles is the stimuli frequency of the part. A blue rectangle represents the part is qualified and detected as IS in the classification, which is a true negative detection. A green rectangle represents the part is defective and detected as defective with the correct frequency, which is a true positive detection. An orange rectangle represents the part is qualified but detected as defective, which is a false positive detection. A red rectangle representing the part is defective but detected as qualified (did not happen in the experiment), which is the false

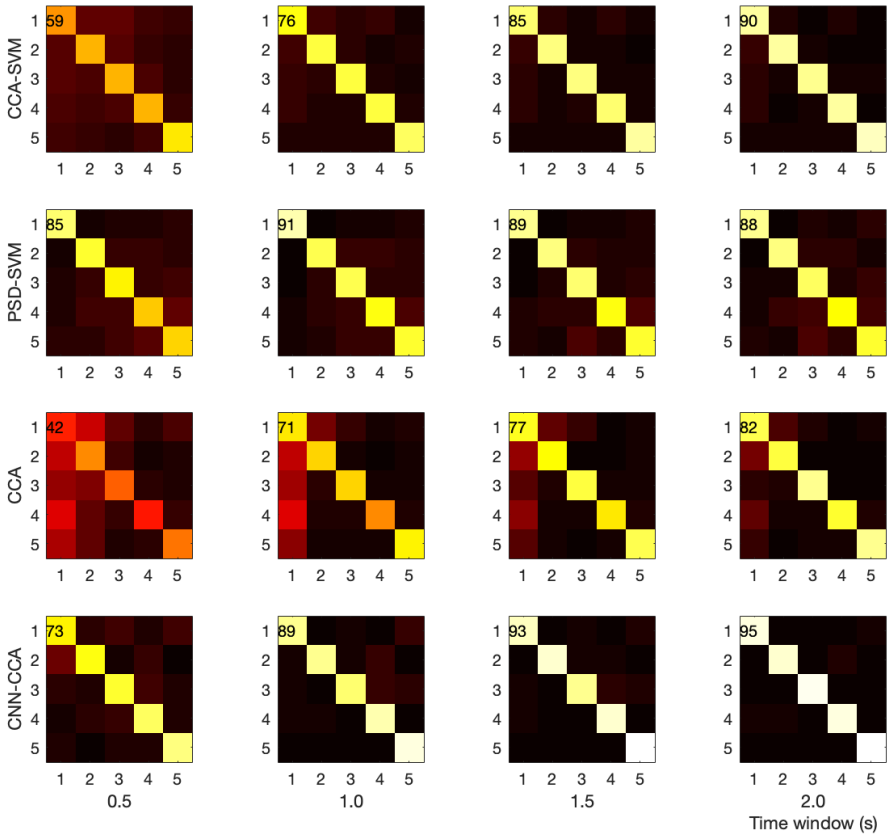


Figure 15: Confusion matrices and IS classification accuracies of CCA-SVM, PSD-SVM, CCA, and CNN-CCA methods at 0.5 s, 1.0 s, 1.5 s, and 2.0 s TW

negative case. The subject 1 and subject 3 completed the experiment with all the part detected correct. The subject 2 got one false positive case in both the first and the second runs. Comparing to their offline experiment, the subject 2 has higher classification accuracy than the subject 3. However, the subject 2 has worse IS classification accuracy, which caused false positive cases.

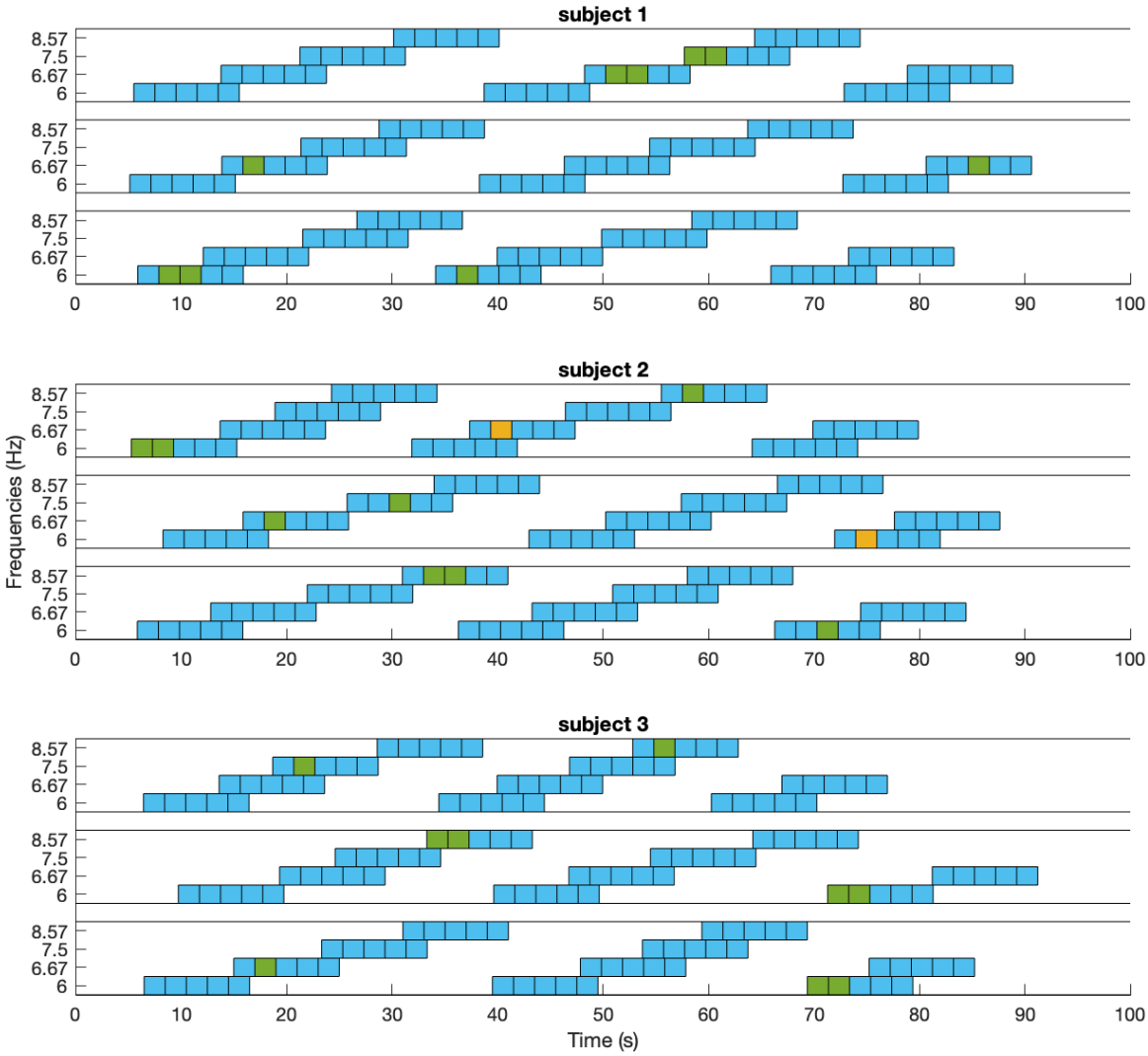


Figure 16: Result of the online experiment

3.9 CONCLUSIONS AND DISCUSSION

We developed a robotic co-worker which obtain the decisions of human collaborators through an EEG-based BCI. The operator can collaborate with the robot in a task without manually operating the robot. We used a manufacturing task, i.e. defective part picking, to demonstrate the concept. The task required the robotic co-worker to select defective parts from the conveyor based on the decisions when human collaborators examine the parts. Human collaborators examine the qualities of the parts from a monitor. The decisions were extracted through a SSVEP-based BCI and sent to the robot.

We established an offline experiment to verify the performance of six SSVEP classification methods, which are CCA, IT-CCA, SVM, CCA-SVM, PSD-SVM, and CNN-CCA. The methods are verified on 0.5 s, 1.0 s, 1.5 s, and 2.0 s TW of EEG data. Our proposed CNN-CCA method achieves best performance among all tested methods for the six tested time window lengths. The average classification accuracies across all five subjects are 80.24%, 89.98%, 93.63%, and 96.56% on 0.5 s, 1.0 s, 1.5 s, and 2.0 s TW, respectively. Then, we established an online experiment with 2.0 s as the SSVEP classification TW. The average part detection successful rate is 93.33%.

In this experiment, we found that combining CNN and CCA can increase the classification accuracy of SSVEP classification. To verify our discovery, we construct a similar CNN and CCA combined model and validated it on a benchmark dataset in the next chapter.

CHAPTER 4 : CONVOLUTION CORRELATION ANALYSIS FOR ENHANCING THE PERFORMANCE OF SSVEP-BASED BCI

4.1 INTRODUCTION

SSVEP is one of the most popular EEG signals for human-machine communication in the field of BCI. Because of the high information transfer rate (ITR), few user training, and ease of trigger, SSVEPs are broadly used in various of applications, such as medical [32,33], industrial [34,35], communication [36,37], smart home [38,39], gaming [40,41,42], and robot & vehicle control [43-47], etc. The performances and ITRs of SSVEP-based BCI applications largely depend on the classification accuracy of the SSVEP signal.

SSVEPs are brain response characterized by frequency pattern at stimulation frequency and its harmonic frequencies [48]. Intuitively, SSVEPs can be detected if transformed into frequency domain. One of the most powerful classification methods is the Power Spectra Density based Analysis (PSDA), which is based on the Discrete Fourier Transform. The PSDA method is designed to be use on the single channel EEG which results the method to be noise sensitive. Moreover, the DFT has low frequency resolution if time window of the EEG is short. To improve the PSDA method, Friman [49] linearly combined EEGs from different channels maximizing signal-noise ratios (SNRs). Meanwhile, Lin [29] came up a spatial filtering approach, i.e. the Canonical Correlation Analysis (CCA). CCA also linearly combines different channels to maximize SNRs, which is similar to Friman's "Maximum Contrast Combination", but it solves

the frequency resolution drawback of the PSDA by calculating the correlations between EEG signal and artificial reference signals in arbitrary frequencies. Currently CCA is still the most popular SSVEP classification method and an important benchmark for new classification methods.

A concern for the CCA method is that its performance varies across different subjects, which means CCA may result low classification accuracy on some subjects. Many researches contribute to improve CCA. One of the most efficient directions is individual calibration data, i.e. constructing subject-specific model for each individual subject based on a training dataset. In CCA, the reference signals are artificial signals constructed with sine and cosine. Artificial signals lack of subject-specific information. Zhang et al. proposed the MwayCCA [50], L1-MCCA [51], and MsetCCA [52] based on standard CCA using subject-specific reference signals constructed from the training dataset for each subject. To further enhance the spatial filter on different harmonic frequency components, Chen et al. [53] built the filter bank CCA (FBCCA) which decompose SSVEPs into multiple sub-band components under multiple pre-processing filters, then fusion the classifications from all sub-band. The training dataset is also utilized to optimize the weights of spatial filters instead of merely being used as reference signals. Wang et al. [54] extended the CCA structure with canonical correlations between training data, validation data, and reference signals. Nakanish et al. [55] introduced ensembled TRCA to optimize spatial filter weights by maximizing the reproducibility in cross-session training dataset. The TRCA achieved an averaged ITR of 325.33 bit per minute in a 40-target cue-guided task. The CORCA-based method proposed by Zhang et al. [56] extracting maximally correlated signal components of EEG from multiple subjects are reported to have better performance than TRCA.

Recently, deep learning techniques have been successfully applied in various classification tasks in many domains. However, in the SSVEP frequency classification task, the performances of deep learning models haven't reach or not even close to simple spatial filters, such as TRCA or ensembled TRCA. Aznan et al. [57] introduced a CNN model in a 5-frequencies SSVEP classification task. They compared the CNN model with other SSVEP deep learning models, such as LSTM, SVM, and RNN, and demonstrated a better performance in their dataset. Similarly, Kwak et al. [58] developed another CNN based model in 5-target SSVEP experiment and obtained higher accuracy than the CCA method. However, these methods are neither investigated on a large number of targets nor validated on a standard benchmark dataset. Podmore [59] et al. proposed a deep CNN framework (PodNet) reached an ITR of 79.26 bit per minute in the same dataset tested for TRCA. PodNet is the first deep learning model which tested on a benchmark dataset and compared with the FBCCA. Except the PodNet has much worse performance than the FBCCA. The main reason that the current deep learning models haven't succeed on SSVEP classification task is overfitting. Because SSVEPs have low signal-to-noise ratios (SNRs), most of the parameters in pure deep neural network frameworks are actually learning from noise. Even with large amount of training data provided, these deep neural networks are still not able to estimate the true model. Since noise is dominating SSVEP signals, classic neural networks, which works well in other domain, can't extract the signal from the training dataset.

Thus, we propose a new deep leaning model for SSVEP classification, called convolution correlation analysis (Conv-CA). The Conv-CA combines CNN structure and traditional correlation analysis. It has two CNN pieces named signal-CNN and reference-CNN. The signal-

CNN provides convolutional operation for the multi-channel EEG signals in every tiny time windows. It transforms EEGs collected from multi-channels into a single signal. Similarly, reference-CNN convolutionally combines reference signals from multiple channels and outputs a one-channel reference signal for each frequency. A correlation layer connecting the signal-CNN and the reference-CNN calculates the correlations between the EEG signal and the reference signal. The CNN structure extends the spatial filter to a non-linear model. Meanwhile, the correlation layer prevents model from overfitting using the correlation analysis. We compared our model with the TRCA method which is one of the state-of-art SSVEP classification model for individual subject.

4.2 BENCHMARK DATASET

Our model Conv-CA and the TRCA [55] are validated on a SSVEP benchmark dataset [60]. The dataset was collected through a cue-guided target selecting task in an offline BCI experiment. EEGs from thirty-five healthy subjects (seventeen females, eighteen males, mean age: twenty-two years) were collected. The experiment included six trials for each subject. Each trial contains 40 simulation tests corresponding to 40 different characters in random order. In each test, subjects were instructed to stare at a target character without eye blinks during the stimulation duration. Each test started with a 0.5-second target cue. Before cue ends, subjects were asked to shift their gaze to the target character. Following the cue, all stimuli flicked on the screen concurrently for 5 seconds. After the 5-seconds simulation, the screen was blanked for 0.5 second before the start of next test.

The 40 target characters of the BCI were arranged in a 5 × 8 matrix on a monitor. The characters were coded in joint frequency and phase modulation (JFPM) approach. The frequencies of the 40 targets were from 8 Hz to 15.8 Hz with a 0.2 Hz offset between two targets. The phase offset between two characters was 0.5 π. EEG data were recorded through a Synamps2 EEG system (Neuroscan, Inc.) under a sampling rate of 1000 Hz. Sixty-four electrodes were placed on the standard positions according to an extended 10-20 system. A reference electrode was placed at the vertex (Cz). During the experiment, electrode impedances were kept below 10 kΩ.

A 50 Hz notch filter was applied to recorded EEGs to remove the power-line noise. All the epochs were down sampled to 250 Hz subsequently. All EEG data were filtered with a Chebyshev Type I filter with cutoff frequencies from 6 Hz to 90 Hz and stopband corner frequencies from 4 Hz to 100 Hz. Because the EEG recording in each test contains 0.5 second pre-stimulus and 0.5 second post-stimulus, we only used the data from 0.5 second to 5.5 seconds for frequency recognition.

4.3 TRCA-BASED METHOD

TRCA is a method that learns the weights of spatial filters from the training dataset of an individual subject. The weights are calculated by maximizing the similarity of EEGs among all trials in the training dataset. Denote the EEG signals as $X \in \mathbb{R}^{N_s \times N_c \times N_f \times N_t}$, where N_s, N_c, N_f, and N_t are the number of sampling points, the number of channels, the number of target SSVEP

frequencies, and the number of trials, respectively. For a frequency target $n \in [1, N_f]$, denote the weight of the corresponding spatial filter as $\boldsymbol{w}_n \in \mathbb{R}^{N_c \times 1}$. The covariance among different trials is

$$\sum_{i,j=1,i\neq j}^{N_t} Cov\left(\boldsymbol{X}^{(*,*,n,i)}\boldsymbol{w}_n, \boldsymbol{X}^{(*,*,n,j)}\boldsymbol{w}_n\right)$$

$$= \sum_{i,j=1,i\neq j}^{N_t} \boldsymbol{w}_n^T \boldsymbol{X}^{(*,*,n,i)^T} \boldsymbol{X}^{(*,*,n,j)} \boldsymbol{w}_n$$

$$= \boldsymbol{w}_n^T \sum_{i,j=1,i\neq j}^{N_t} \boldsymbol{X}^{(*,*,n,i)^T} \boldsymbol{X}^{(*,*,n,j)}\, \boldsymbol{w}_n$$

$$= \boldsymbol{w}_n^T \boldsymbol{S} \boldsymbol{w}_n \qquad (4.1)$$

Here, the $\boldsymbol{X}^{(*,*,n,i)}$ and $\boldsymbol{X}^{(*,*,n,j)}$ denote the n-th frequency in the i-th and j-th trials of $\boldsymbol{X}$.

The covariance among all the trials is

$$\sum_{i,j=1}^{N_t} Cov\left(\boldsymbol{X}^{(*,*,n,i)}\boldsymbol{w}_n, \boldsymbol{X}^{(*,*,n,j)}\boldsymbol{w}_n\right)$$

$$= \boldsymbol{w}_n^T \boldsymbol{Q} \boldsymbol{w}_n \qquad (4.2)$$

Note that the equation (4.2) has different form as represented in [55], but they are mathematically equivalent.

The weight $\boldsymbol{w}_n$ of the spatial filter for the target n can be calculated by maximizing the covariance among different trials divided by the covariance among all trials, i.e.

$$\boldsymbol{w}_n^* = \underset{\boldsymbol{w}_n}{\operatorname{argmax}} \frac{\boldsymbol{w}_n^T \boldsymbol{S} \boldsymbol{w}_n}{\boldsymbol{w}_n^T \boldsymbol{Q} \boldsymbol{w}_n} \tag{4.3}$$

The optimal weight $\boldsymbol{w}_n^*$ can be calculated as the eigenvector of matrix $\boldsymbol{Q}^{-1}\boldsymbol{S}$ corresponding to the largest eigenvalue. We denote the above steps for obtain optimal $\boldsymbol{w}_n^*$ as a function

$$\boldsymbol{w}_n^* = \gamma(\boldsymbol{X}, n) \tag{4.4}$$

For an individual subject with a training dataset $\boldsymbol{X}_{train} \in \mathbb{R}^{N_s \times N_c \times N_f \times N_t}$ and test data $\boldsymbol{X}_{test} \in \mathbb{R}^{N_s \times N_c \times N_f}$, the reference signal $\boldsymbol{Y} \in \mathbb{R}^{N_s \times N_c \times N_f}$ is the trial-averaged training data, i.e.

$$\boldsymbol{Y}^{(*,*,n)} = \frac{1}{N_t} \sum_{i=1}^{N_t} \boldsymbol{X}_{train}^{(*,*,n,i)}, \quad n \in [1, N_f] \tag{4.5}$$

The correlation coefficients between the test data $\boldsymbol{X}_{test}$ and its reference signal $\boldsymbol{Y}$ are

$$\lambda_n = \rho\left(\boldsymbol{X}_{test}^{(*,*,n)} \boldsymbol{w}_n^*, \boldsymbol{Y}^{(*,*,n)} \boldsymbol{w}_n^*\right), \quad n \in [1, N_f] \tag{4.6}$$

where $\boldsymbol{w}_n^* = \gamma(\boldsymbol{X}_{train}, n)$ and $\rho(.,.)$ indicates the one-dimensional correlation analysis. Then, the frequency of the test data $\boldsymbol{X}_{test}$ can be recognized by

$$f_n, \quad n = \operatorname{argmax} \lambda_n, \quad n \in [1, N_f] \tag{4.7}$$

4.3.1 Ensemble TRCA-based method

Experiment results shows that integrating all spatial filters of all frequency targets could further improve the performance of TRCA. With N_f spatial filters, a concatenated matrix $\boldsymbol{W}^* \in \mathbb{R}^{N_c \times N_f}$ can be obtained. Then the equation (4.6) can be replaced by

$$\lambda_n = \rho\left(\boldsymbol{X}_{test}^{(*,*,n)}\boldsymbol{W}^*, \boldsymbol{Y}^{(*,*,n)}\boldsymbol{W}^*\right), \quad n \in [1, N_f] \tag{4.8}$$

where $\rho(.,.)$ indicates the two-dimensional correlation analysis instead.

4.3.2 TRCA with filter bank

Filter bank technique have been verified to improve the performance of TRCA-based method in [55]. The filter bank analysis decomposes SSVEP signals into sub-band components using infinite impulse response (IIR) filters, such that the harmonic components of SSVEP signals can be embedded independently. According to the study [53], the lower and upper cut-off frequencies of the m-*th* sub-band were set to $m \times 8$ Hz and 90 Hz, where $m \in [1, N_b]$ and N_b is the number of sub-bands. By applying the m-*th* Chebyshev Type I filters to the training dataset $\boldsymbol{X}_{train}$ and test data $\boldsymbol{X}_{test}$, a sub-band correlation for the frequency target n can be obtained as λ_n^m from equation (4.6) or equation (4.8). With all sub-bands, a set of correlations $\boldsymbol{\beta}_n = \left(\lambda_n^1, \lambda_n^2, \ldots, \lambda_n^{N_b}\right)^T$ for the target n can be linear combined by

$$\lambda_n = \boldsymbol{\Phi}^T\boldsymbol{\beta}_n \tag{4.9}$$

where $\boldsymbol{\Phi}$ is defined as

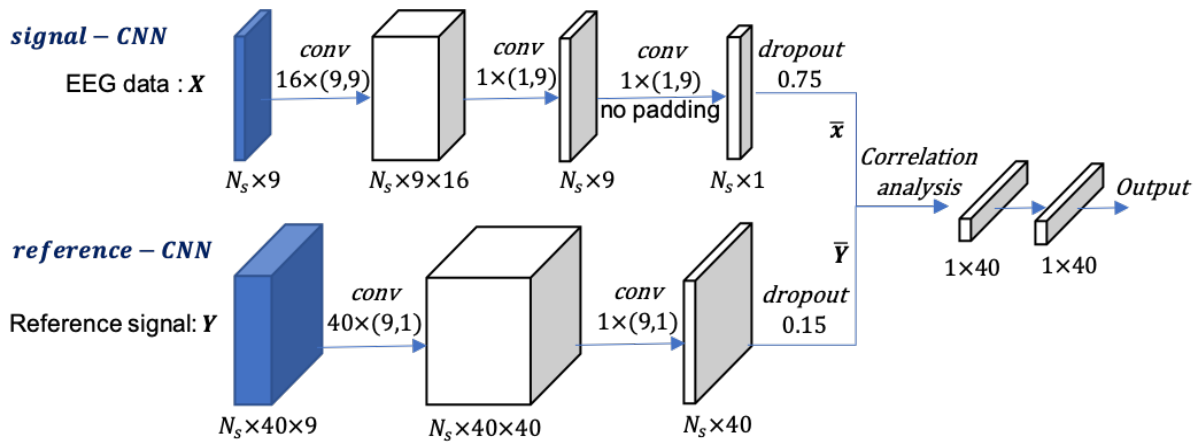


Figure 17: Structure of the proposed Conv-CA SSVEP classification model. The CNN layers use zero paddings except the second layer of the signal-CNN which has no padding. All the CNNs use linear activation functions. The activation function of the dense layer is softmax.

$$\boldsymbol{\Phi} = (1^{-a} + b \quad 2^{-a} + b \quad \dots \quad {N_b}^{-a} + b)^T \tag{4.10}$$

Here a and b are empirically set to 1.25 and 0.25 respectively.

4.4 CONV-CA MODEL DESIGN

The Conv-CA is designed to classify multi-target SSVEP signals with a structure combining CNN and correlation analysis. The detailed structure is demonstrated in Figure 17. The Conv-CA takes the EEG data $\boldsymbol{X} \in \mathbb{R}^{N_s \times N_c \times 1}$ and the reference data $\boldsymbol{Y} \in \mathbb{R}^{N_s \times N_c \times N_f}$ as inputs. To simplify the implementation, we rearranged the order of dimensions of $\boldsymbol{Y}$ from $\mathbb{R}^{N_s \times N_c \times N_f}$ to $\mathbb{R}^{N_s \times N_f \times N_c}$. Then, we apply a 3-layer CNN to $\boldsymbol{X}$ as

$$\bar{\boldsymbol{x}} = f(\boldsymbol{X}), \quad \bar{\boldsymbol{x}} \in \mathbb{R}^{N_s \times 1} \tag{4.11}$$

Here the 3-layer CNN $f(.)$ is named as signal-CNN. Another 2-layers CNN is applied to the reference signal $\boldsymbol{Y}$ as

$$\overline{\boldsymbol{Y}} = g(\boldsymbol{Y}), \quad \overline{\boldsymbol{Y}} \in \mathbb{R}^{N_s \times N_f} \tag{4.12}$$

where $g(.)$ is named as reference-CNN. The correlation analysis layer takes $\overline{\boldsymbol{x}}$ and $\overline{\boldsymbol{Y}}$ as inputs and calculates the correlation coefficients of $\overline{\boldsymbol{x}}$ and $\overline{\boldsymbol{Y}}^{(.,n)}$ for all $n \in [1, N_f]$. By denoting $\overline{\boldsymbol{y}}_n = \overline{\boldsymbol{Y}}^{(.,\boldsymbol{n})}$, the correlation coefficients in the correlation analysis layer are

$$\gamma(\overline{\boldsymbol{x}}, \overline{\boldsymbol{y}}_{\boldsymbol{n}}) = \frac{\sqrt{(\overline{\boldsymbol{x}} \circ \overline{\boldsymbol{x}})^T (\overline{\boldsymbol{y}}_{\boldsymbol{n}} \circ \overline{\boldsymbol{y}}_{\boldsymbol{n}})}}{\sqrt{\overline{\boldsymbol{x}}^T \overline{\boldsymbol{x}}} \sqrt{\overline{\boldsymbol{y}}_n^T \overline{\boldsymbol{y}}_n}}, \quad n \in [1, N_f] \tag{4.13}$$

where $\circ$ is the Hadamard product (i.e. element-wise product). Then the output of the correlation analysis layer is

$$\begin{aligned} \boldsymbol{z} &= h(\overline{\boldsymbol{x}}, \overline{\boldsymbol{Y}}) \\ &= \left[\gamma(\overline{\boldsymbol{x}}, \overline{\boldsymbol{y}}_1) \quad \gamma(\overline{\boldsymbol{x}}, \overline{\boldsymbol{y}}_2) \quad \ldots \quad \gamma\left(\overline{\boldsymbol{x}}, \overline{\boldsymbol{y}}_{N_f}\right)\right] \in \mathbb{R}^{N_f} \end{aligned} \tag{4.14}$$

We use a dense layer with N_f units and softmax activation function as the final layer before classification.

Because we build a correlation analysis layer at the end of CNNs, the CNNs do not have to be active by non-linear activation functions. In our implementation, we use linear activation functions in all CNNs. The first layer of the signal-CNN has 16 filters of 9×9 kernels. It convolutes EEGs in all input channels ($N_c = 9$) in a short local time period (9 sampling points or 36 ms). The second layer combines the 16 filters in the first layer into one. It uses 1×9 kernels to

weight EEGs from different channels. The third layer applies a 1×9 kernel with no padding (The first and second layers use zero paddings to keep the output to have same dimension as the input.) to transform the data $\boldsymbol{X} \in \mathbb{R}^{N_s \times N_c \times 1}$ into a one-dimension signal $\overline{\boldsymbol{x}} \in \mathbb{R}^{N_s \times 1}$. At the end of signal-CNN, we apply a dropout with dropping rate 75% to $\overline{\boldsymbol{x}}$ for regularization. The reference-CNN uses two CNN layers. The first layer contains 40 ($N_f = 40$) filters of 9×1 kernels. It provides convolutional weights for all classes and all channel EEGs in the short time period. The second layer uses 9×1 kernels to integrate the weights. Similar as the signal-CNN, a dropout layer with dropping rate 15% is added at the end of the reference-CNN.

The Conv-CA is implemented in python-Keras with tensorflow backend. We use categorical cross-entropy as loss function. The optimization is solved with Adam algorithm (learning rate (8e-4), beta1 (0.9), beta2 (0.999)) with a batch size 32. The model runs on the Google Cloud platform.

4.5 PERFORMANCE EVALUATION

Accuracy of Classification and ITR are used to evaluate the performance of all models. While calculating ITRs, the 0.5 second gaze-shifting time from the original off-line experiment is added. We evaluate all the methods in various data lengths (0.2 s to 1 s, with a step of 0.2 s). The data is extracted from the simulation period (0.5 s to 5.5 s) of the 6 s recorded data with a step of $0.15 \times data\ length$. TRCA-based methods are used to compare with our proposed Conv-CA model. We use leave-one-out cross validation to evaluate the methods. Specifically, one of the 6

trials of the EEG data are used as test data and the other 5 trials are used as training dataset. This process is repeated for 6 times so that every trial is tested. Paired t-test are performed to verify the statistical differences of the Conv-CA model and TRCA-based methods.

The Conv-CA model is trained with a Tesla K80 GPU. When the time window $N_s = 50$ (0.2 s), each epoch takes 11 s. The model converges within 100 epochs. To avoid gradient exploding, a gradient clipping at 5.0 is added in the Adam solver. For $N_s = 100$ (0.4 s), $N_s = 150$ (0.6 s) and 200 (0.8 s), we use 500 epochs while each epoch takes 9 s, 8 s and 7 s respectively. As for $N_s = 250$ (1.0 s), 1,000 epochs are applied. Each epoch takes 7 s. We demonstrate an example of the epoch selection in Figure 18. Figure 18 shows the training loss, validation loss, training accuracy, and validation accuracy while applying Conv-CA on the 6-th trial EEG data of subject 35 with $N_s = 250$. In Figure 18. (a), the training accuracy and the validation accuracy increase rapidly in the first 500 epochs. Then they slowly grow and converge in the rest 500 epochs. In Figure 18. (b), the training loss decreases over the entire 1,000 epochs. However, the validation loss almost stops decreasing in the last 500 epochs. The model converges within 1,000 training epochs. Similar training loss and validation loss ensure the model have similar performance on both the training dataset and the test data.

The Conv-CA is compared with two TRCA-based methods, i.e. basic TRCA and ensemble TRCA. According to [55], the basic TRCA and the ensemble TRCA achieve best performance with 5 sub-bands. Thus, we applied filter bank technique on both methods with 5 sub-bands.

4.6 RESULT

We investigated the performance of our proposed Conv-CA model by comparing with TRCA-based methods. Because, as mentioned above, filter bank technique (5 sub-bands) can substantially improve the classification accuracy of TRCA-based methods, we only explored the basic TRCA method under 5 sub-bands and the ensemble TRCA method under 5 sub-bands. For simplification, we will mention them as TRCA and ensemble TRCA in below. Figure 19 shows the classification accuracy and ITR comparisons across all subjects in various time windows. By Figure 19 (a) and (b), our proposed Conv-CA model significantly outperform ($p < 0.001$) the basic TRCA method in both classification accuracies and ITRs across 5 tested time windows from 0.2 s to 1.0 s. Figure 19 (c) and (d) demonstrate that the Conv-CA model is significantly ($p < 0.05$) better than the ensemble TRCA method in both accuracies and ITRs across all time windows. The Conv-CA model achieves the highest ITR 226.19 bits/min on the 0.4 s time window, which

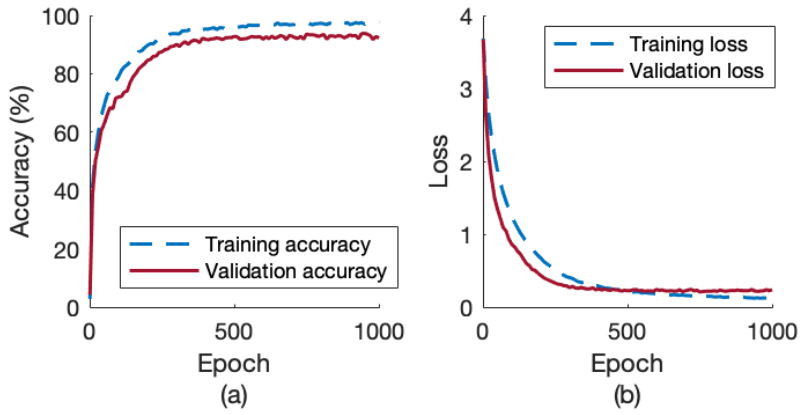


Figure 18: (a) The training accuracy, the validation accuracy, (b) the training loss, and the validation loss of the 6-th trial EEG of subject 35 on 1.0 s time window over 1,000 training epochs.

is higher than the highest ITR 216.34 bits/min of the ensemble TRCA achieved on the same time window. The highest ITR of the Conv-CA is significantly ($p < 0.001$) higher than the highest ITR of the ensemble TRCA method.

We also investigated how our proposed Conv-CA model performs on those subjects which have relative lower classification accuracies under TRCA-based methods. In Figure 20, we plotted the absolute performance increment of Conv-CA compare to the ensemble TRCA vs the classification accuracy of the ensemble TRCA. The classification accuracies of the subjects, whose

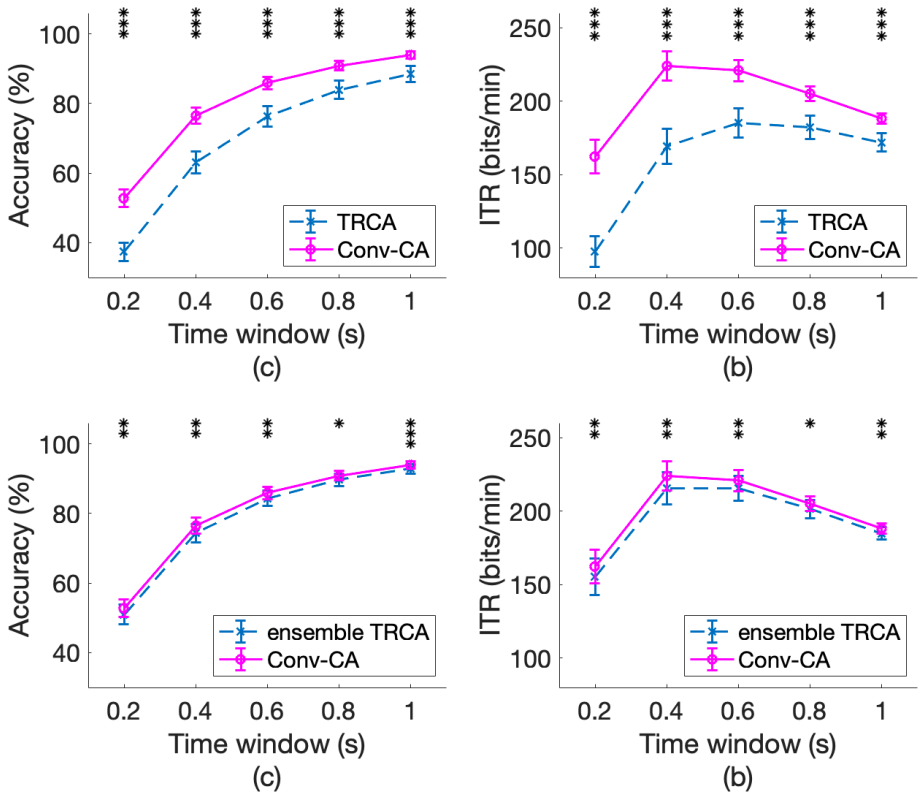


Figure 19: The averaged accuracies (a, c) and ITRs (b, d) across all subjects by TRCA-based methods and Conv-CA model in various time windows. The asterisks in subfigures indicate significant difference between two methods by paired t-tests (*$p < 0.05$, ** $p < 0.01$, *** $p < 0.001$). The error bars indicate standard errors.

SSVEP were not well-classified by the ensemble TRCA, increased much more than those subjects whose SSVEP were well-classified. Especially in the 0.8 s time window, the classification accuracy of subject 33 increased from 43.57% to 61.71% with the Conv-CA model. And the accuracy of subject 19 increased from 58.93% to 71.57%. According to Figure 20, the Conv-CA tends to have better improvement for low-performed subjects in the ensemble TRCA method. To verify the assumption, we calculated the correlation coefficients of the increments and the classification accuracies of the ensemble TRCA. The hypothesis tests and correlation coefficients in Figure 21 indicate that the increments have negative relationship with the accuracies of the ensemble TRCA. Thus, our proposed Conv-CA model has better performance on the subjects whose SSVEP have lower- performance in ensemble TRCA method.

4.7 CONCLUSION AND DISCUSSIONS

Enhancing the classification performance of SSVEPs is important for numbers of BCI applications. In this study, the Conv-CA model we proposed has better performance than the state-of-art TRCA-based methods on both classification accuracies and ITRs for classifying multi-target SSVEPs using individual subject data. The TRCA-based methods learn weights of spatial filters

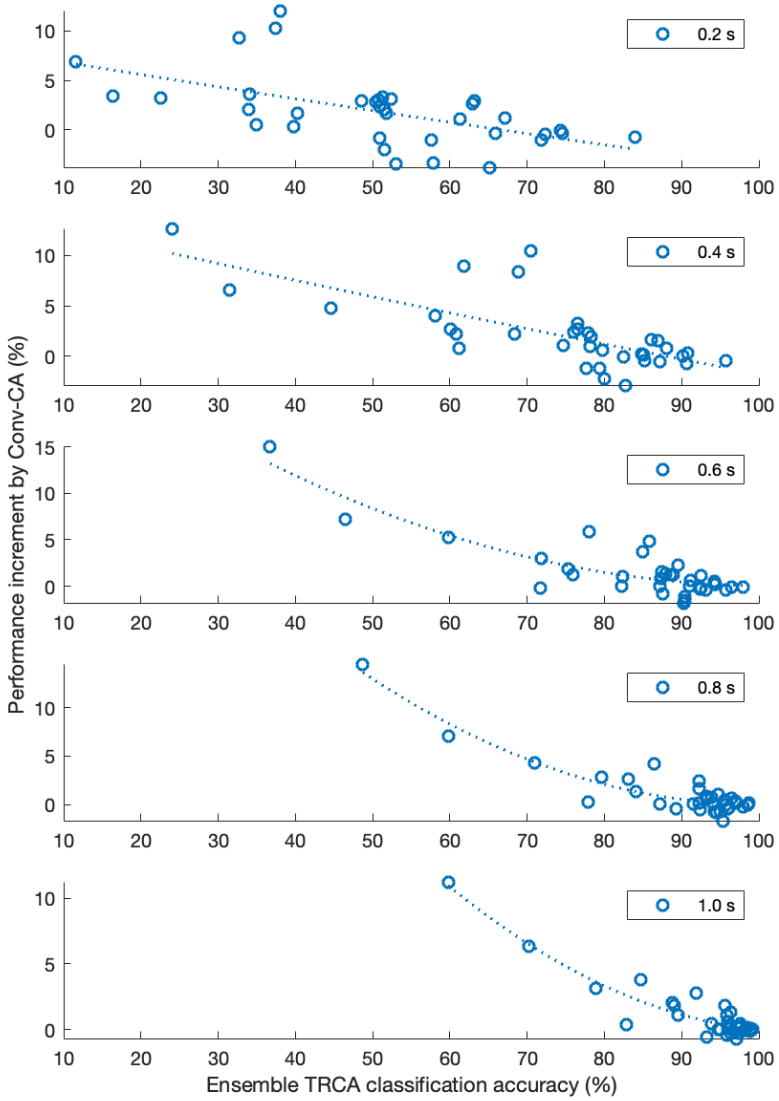


Figure 20: The Conv-CA performance increments from ensemble TRCA vs the classification accuracies of ensemble TRCA method in various time windows. The dash lines are 2-nd order polynomial curve fittings of the increments to visualize the trends.

the ensemble TRCA with filter bank, the method has great performance on the benchmark dataset. However, for some subjects, for example the subject 11, 19, and 33, the classification accuracy is much lower than the cross-subject averaged accuracy. Specifically, the accuracies of the subject 11, 19, and 33 are 76.20%, 69.63%, and 57.78% compare to the averaged accuracy 92.99% on the 1.0 s time window. Our proposed Conv-CA model has significant better performance on these low-performed subjects.

The proposed Conv-CA model no longer using linear models, such as spatial filters. It combines CNNs and correlation analysis to provide convolute operations for short periods of EEG signals across multiply channels. The Conv-CA suggests that by involving non-linear components, we can obtain a better performed model than spatial filters. It also implies that some subjects have worse SSVEP performances is probably because the model does not contain enough non-linear

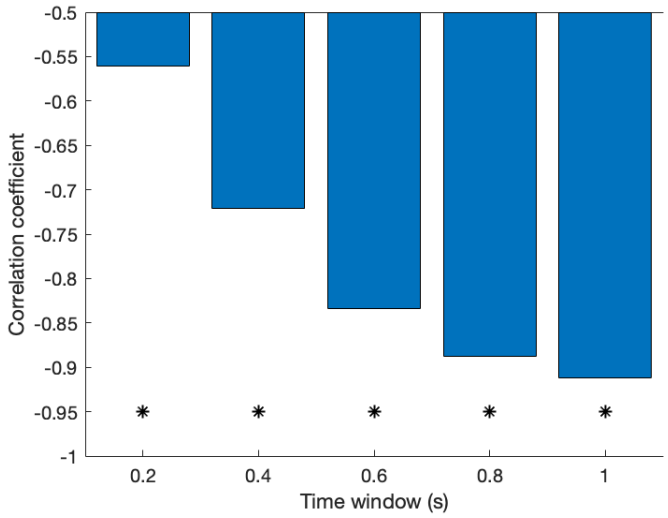


Figure 21: Correlation coefficient of the Conv-CA increments from ensemble TRCA and the accuracies of ensemble TRCA in various time windows. The * indicates significance of a correlation ($p < 0.001$).

components. Furthermore, the CNN structures and parameters in the Conv-CA model has not been sufficiently explored. By constructing different CNNs, the Conv-CA can expand into infinite number of models. So, there is great potential that a fine-tuned Conv-CA model can achieve much better performance than the current Conv-CA model. And the structure of the Conv-CA can also be further changed to the combinations of other non-linear structures and other signal processing tools.

The Conv-CA model is not a classic deep learning model which constructed by pure neural networks. It contains a correlation analysis layer at the end of the CNNs. The correlation analysis layer connects the signal-CNN and the reference-CNN by correlation coefficients. In other aspect, the CNNs are extensions of spatial filters, because the signals and the references are still connected with correlation analysis. Classic deep learning models learn all parameters from the dataset. Because EEG signals has low SNRs, it is easy to obtain a biased deep learning model, which means the model learns too much from the noise rather than focus on the signal. Even with huge amount of data, it will still be a challenge for deep learning models to locate the true signal. By involving a correlation analysis layer, the CNNs are forced to maximize the signal-reference correlations. It can avoid bring too much redundant parameters and structures.

The idea of combing traditional signal processing tools and deep learning structures can also be tested on other low SNR signals, such as P300 and seizure. Deep learning structures can bring sufficient non-linear operations to a model and the unsupervised features form traditional signal processing tools can prevent overfitting caused by involving too many parameters.

CHAPTER 5 : EEG-BASED FEAR DETECTION FOR HUMAN SAFETY PROTECTION IN HUMAN-ROBOT COLLABORATION

5.1 INTRODUCTION

In human-robot collaborative interaction, the robot collaborators should adapt their actions to human's physical and mental states. For example, the robot is expected to slow down when the human collaborator is exhausted. Or when the collaborator feels uncomfortable, the robot should stop. One of the important features is that the robot should slow down, pause, or stop when the human feels unsafe in the collaboration. These functions require the robot to understand the emotions of humans.

Emotions can be acquired indirectly from the human's facial expression, body language, and voice by robots [61-64]. They can also be perceived from the physiological signals, such as electroencephalogram (EEG), electromyogram (EMG), and electrocardiogram (ECG) [65-67]. EEG is the most commonly used physiological signal for emotion recognition. It is collected from the brain cortex hence directly reflects the mental activities. However, emotion recognition from EEG is challenging, because of the emotion-variety in different individuals and the fuzzy boundaries of the emotional time, etc. Most current researches stimulate subjects to produce certain emotions by showing pictures, sounds, music, and videos with emotional tendencies [66,68,69]. Stimulated emotions are disgust, joy, anger, sadness, surprise, fear, curiosity, etc. And the EEG time window for emotion recognition varies from around 10 seconds to couple minutes. However,

in the human-robot collaboration scenario, the robot should be able to respond to emotions quickly, especially fear. Moreover, emotions produced by watching videos or images are different from the emotions produced while working collaboratively with a robot. But there is no study detecting the emotions while subjects are working with a robot.

In this study, we detected the fear response from EEG in real-time (1 second time window) while the humans are working with a robot collaboratively. A human subject experiment was designed to collect the fear responses that human-produced when they observe unexpected robot motions. Classification models and regression models for detecting fear responses in EEG data per second were built and compared based on classification accuracies or prediction errors. Finally, the models were evaluated based on their performance for unsafe warning.

5.2 EXPERIMENTAL DESIGN

We designed an experiment to collect the human operators' EEG responses when they felt dangerous while working collaboratively with the robot. The experiment design followed two requirements: 1) Subjects need to feel unsafe; 2) The robot will not harm subjects physically. Therefore, in our experiment, subjects were asked to stay outside the workspace of the robot. Meanwhile, we enhanced their intense feeling from both visual and auditory aspects. The experiment was designed into two stages. In the first stage, subjects were assigned a task so that they always focused on the robot. At random times, the robot jumped into the second stage. Since stress can modulate sensory gating [70, 71], the unexpected stage change while subjects were

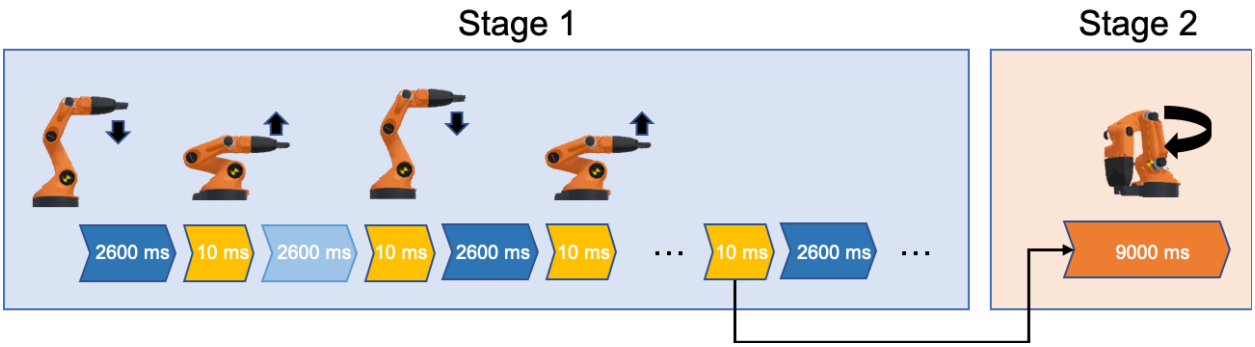


Figure 22: Illustration diagram of the robot motion in the experiment design. Stage 1: the robot moves down for 2600 ms, stays at the lowest point for 10 ms, then, moves up for 2600 ms, and stays at the highest point for another 10 ms. The downward or upward motion repeats for 30 times. But a random time between the 21-st motion and the last motion, the robot can enter the stage 2. Stage 2: the robot moves quickly in a predesigned motion for 9000 ms.

focusing on the task can enhance the fear of the subjects. In the second stage, the robot moved quickly under unpredictable trajectories and simultaneously destroyed surrounding objects. Scenes and sounds that destroying objects also enhanced the sense of unsafe. The robot actions and destructions in the second stage were all pre-designed to ensure the safety of subjects. The EEG was collected during both stages.

The experimental design details are as follows. The subject sits one meter away from the robot with a keyboard accessible by them. Before the experiment, the subject was told to complete a task, which is during the robot moving back and forth, press a key when the robot moves to the highest position or the lowest position. As the experiment starts, the robot end-effector slowly moves downward as shown in the two leftmost robot configurations in Figure 22. The downward movement takes 2600 ms. Then, the robot pauses for 10 ms at the lowest position. The subject needs to press a key so that the key hits in the 10 ms pause period of the robot. If the key is successfully pressed within the pause time, the subject will receive a point for the task. Otherwise,

a sound message will tell the subject whether the key was pressed too early or too late. Then, the robot moves up and pauses for another 10 ms. And the subject should once again press the key within the pause time. The upward or downward motions repeat 30 times in total. However, if the subject scored more than 7 points in the first 20 motions, the experiment will leap to the second stage at a random time between the 21-*st* motion and the last motion. Once the experiment enters the second stage, the robot will move quickly following a pre-designed trajectory. However, the trajectory appears to be unpredictable to the subject. During the quick motion, the robot pushes a large box and two bolts off the table. The subject will hear the dropping sound of bolts. Then, the robot will break the pen attached at the end of the robot. The whole quick motion takes 9000 ms. The EEG is collected during the entire experiment. The time when the experiment enters the second stage is also recorded.

5.3 EEG COLLECTION

The EEG was collected through a B-Alert X24 headset (Advanced Brain Monitoring, Carlsbad, CA), which has 20 electrodes positioned following the 10-20 system and a pair of reference channels. The sampling frequency is 256 Hz. Five healthy subjects (males, age 25-30 years) participated in the experiment. The experiment run 4 times for each subject.

5.4 SIGNAL EXTRACTION

The fear acquisition process can activate several brain areas. Kevin S. LaBar et al. conducted a study using echoplanar functional magnetic resonance imaging (fMRI) to observe the fear acquisition in 1998 [72]. The experiment is designed in a simple discrimination paradigm. A visual conditioned stimulus was paired with an electric shock to the wrist, while the other visual conditioned stimulus was presented alone. In the study, activations were observed in the anterior cingulate (Brodmann's Area (BA) 32, 24), precentral gyrus (BA 6), periamygdaloid cortex (BA 34), superior frontal gyrus (BA 6), precentral gyrus (BA 4), middle frontal gyrus (BA 8), superior temporal gyrus (BA 22), and striatum. Another two studies conducted by M. A. Fullana et al [73] and Jorge L. Armony [74] confirmed the activations in the anterior cingulate, superior frontal gyrus, middle frontal gyrus, and superior temporal gyrus. They also observed additional activations in the fusiform gyrus (BA 37) and dorsolateral prefrontal cortex (BA 9, 46). We marked all the reported activation areas on Figure 23 (a). Now that we are observing fear from the EGG. The corresponding channels in a 20-electrodes 10-20 system to the activation areas are Fp1, Fp2, F3, F4, T3, and T4, as shown in Figure. 23 (b). In the study conducted by Michael A. Kisley [75], they found event-related gamma (30-100 Hz) and beta (14-30 Hz) activities in the EEGs after a click which indicates that the participants will experience white noise burst or electrical shock.

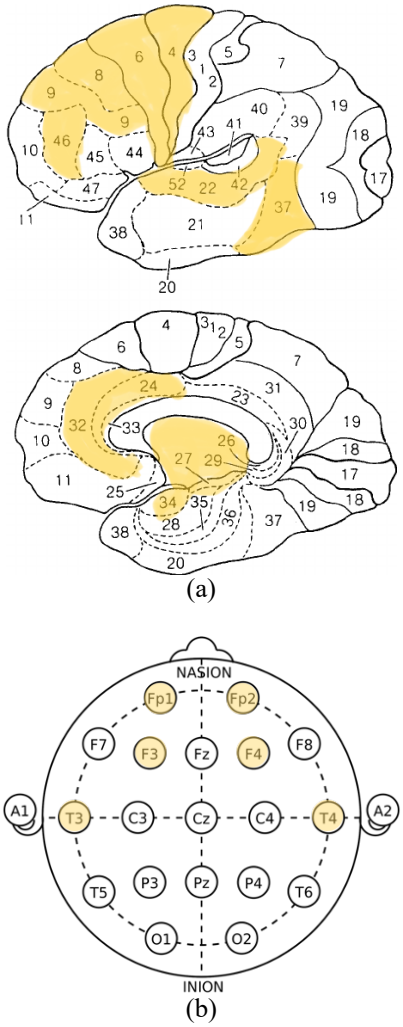


Figure 23: (a) The brain active areas in the process of fear acquisition in fMRI studies and (b) the corresponding positions in the EEG 10-20 system. The numbers in (a) are Brodmann's areas.

These relative studies suggested that fear responses can be found in gamma and beta bands of the Fp1, Fp2, F3, F4, T3, and T4 channels. We have consistent discovery in our experimental study. In Figure 24, we plotted the average power spectral density (PSD) of the EEG from 16 Hz to 45 Hz collected from the Fp1, Fp2, F3, F4, T3, and T4 channels of all subjects. The average PSD increased after the experiment enters the stage 2.

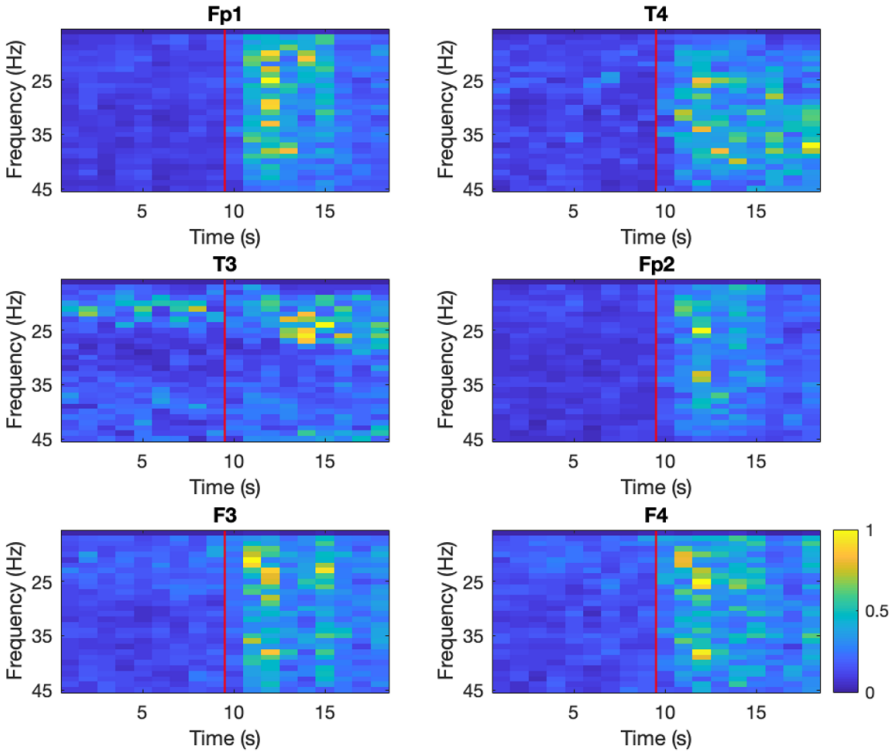


Figure 24: The average PSD of EEG from 16 Hz to 45 Hz collected from Fp1, Fp2, F3, F4, T3, and T4 channels of all subjects. The red line indicates the time when the experiment entered stage 2. The left and right sides are the PSD of 9 s EEG before and after the stage change. PSD is calculated in non-overlapping 1 s time windows and normalized between 0 and 1.

5.5 SIGNAL PREPARATION

The fear response is to alert the robot when human feels unsafe in the collaboration. Such that, the robot can interrupt its current motion by stopping or slowing down to protect human safety. Under this goal, the detection was performed in every second. Detection models were built for each individual subject. The data of each individual subject is 4-folded and divided into a training dataset, a validation dataset, and a test dataset. Specifically, every 3 runs of the experiment were used for training. The rest run was left for validation and test. The training and validation

datasets are the EEG segments collected from 9 seconds before and after the experiment stage change. The test dataset is the EEG collected from 50 seconds before and 9 seconds after the stage change. The EEG segments have no overlapping.

5.6 MODEL EVALUATION METRIC

The ideal detection model should classify all the 1-second EEG segments collected in the experiment stage 1 as negative for fear response and classify at least one segment collected in the stage 2 as positive for fear response. Because the robot will take actions to protect human safety once one EEG segment is classified as positive fear response, some EEG segments collected in stage 2 are allowed to be classified as negative. Models are evaluated using Fowlkes–Mallows index (FM) as

$$FM = \sqrt{\frac{TP}{TP + FP} \cdot \frac{TP}{TP + FN}} \qquad (5.1)$$

Here, TP is the number of true positives, specified as the number of test trials which has at least one positive fear response detected in the experiment stage 2. The FN is the number of false negatives. It is the number of all test trials minus TP. FP is the number of false positives, which is the total number of positive fear responses detected in the experiment stage 1.

5.7 FEAR RESPONSE CLASSIFICATION

PSD from 16 Hz to 45 Hz of the EEG in different channel combinations were used as features. The channel combinations were (Fp1, Fp2), (Fp1, Fp2, T3, T4), (Fp1, Fp2, F3, F4), (T3, T4, F3, F4), and (Fp1, Fp2, T3, T4, F3, F4). The EEG segments collected in the experiment stage 1 were labeled as 0. And segments collected in the stage 2 were labeled as 1. EEGs were classified by logistic regression (LGS) and support vector machine (SVC). The LGS model was applied with L1-regularization and the SVC model was applied with RBF-kernel.

We chose hyperparameters which maximize the score of the validation dataset. For the fear detection task, we tested two different validation scores, i.e. classification accuracy (CA) and a modified classification accuracy (MCA). CA is the ratio of correct classification. While for the MCA, it counted all the false negatives (i.e. negative fear response in stage 2) as true positives (positive fear response in stage 2) if there was at least one true positive found in a test trial. Comparing to CA, MCA is more similar to the model evaluation metric.

5.8 FEAR RESPONSE REGRESSION

Because it is not necessary for every EEG segment in the experiment stage 2 to be classified as positive fear response. The fear response detection can be performed by regression models. For the regression models, we used the same features and channel combinations as in classification.

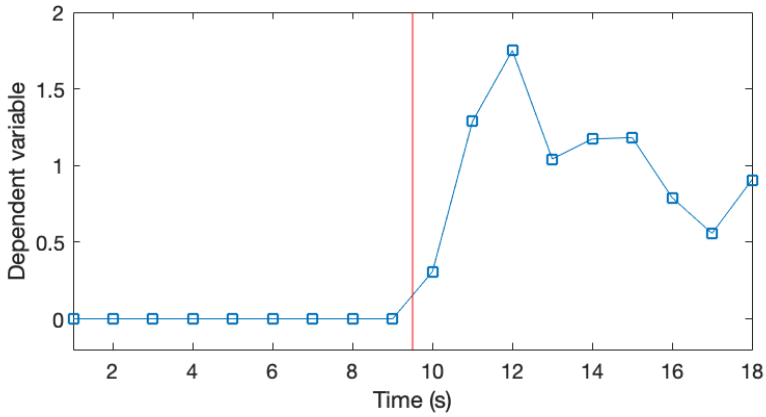


Figure 25: Dependent variable for regression models

The dependent variables of EEG segments in the stage 1 were set to 0. The dependent variables in the stage 2 at the t-*th* second after the stage change was set as

$$y^{(t)} = \frac{\bar{p}_2^{(t)} - mean(\bar{p}_1^{(i)})_{i=1,\dots,T}}{mean(\bar{p}_2^{(i)})_{i=1,\dots,T} - mean(\bar{p}_1^{(i)})_{i=1,\dots,T}} \quad (5.2)$$

where, $\bar{p}_1^{(t)}, \bar{p}_2^{(t)} \in \mathbb{R}$ are the average PSD of the t-th second EEG in the stage 1 and 2 from 16 Hz to 45 Hz in the channels Fp1, Fp2, T3, T4, F3, and F4 across all experiment runs. The dependent variables of all the time windows is shown in Figure 25.

We applied ridge (RDG), lasso (LSS), and linear support vector regression (SVR) models for regression. The validation loss is a balanced mean absolute error (MAE). The balanced MAE is the average of the MAE of the prediction for stage 1 and the MAE of the prediction for stage 2. The regression prediction is further classified as a positive fear response if it satisfies

$$\hat{y}^{(t)} > c + mean(\hat{y}^{(i)})_{i=1,\dots,t} + 2 * std(\hat{y}^{(i)})_{=1,\dots,t} \quad (5.3)$$

Here, c is a constant threshold to control the detection sensitivity. $\hat{y}^{(t)}$ is the regression prediction of the t-*th* second EEG segment. $std(\hat{y}^{(i)})_{=1,\dots,t}$ is the standard deviation of the predictions in the first *t* seconds.

5.9 RESULTS

Table 2 shows the accuracy and model performance comparison t-test when validating SVC and LGD on various channel combinations using classification accuracy and modified classification. Applying SVC to the Fp1, Fp2, F3, and F4 channels (SVC-front4) reached the highest accuracy for both validation scores at 86.0% and 96.0%. While applying SVC merely to the Fp1 and Fp2 channels (SVC-front2) reached the second-best accuracy for both validation scores at 85.8% and 95.7%. These two models have significantly better performance than other tested models.

Figure 26 (a) and (b) show the classification results of SVC-front4 and SVC-front2 on the test dataset. SVC-front4 obtained 100% true positive rates. However, there were 5.05 false positives on average per trial. SVC-front2 also got 100% true positive rates with lower false positives rates as 4.00 false positives per trial.

Table 3 shows the modified MAE of RDG, LSS, and SVR on various channel combinations. Applying SVR on the Fp1, Fp2, F3, and F4 channels (SVR-front4) reached the lowest MAE at 0.372. SVR-front4 has significant lower MAE than the other tested models except

for the SVR-front2 model which applied SVR on the Fp1 and Fp2 channels. SVR-front2 obtained higher MAE at 0.383.

Figure 26 (c) and (d) show the classification results of SVR-front4 and SVR-front2 with threshold c=0.2. The true positive rates are 95% and 80% for SVR-front4 and SVR-front2 respectively. However, their false positive rates are much lower than the classification models as 0.20 and 0.1 false positives per trial respectively.

When c=0.2, FM of SVR-front4 is 0.886 and FM of SVR-front2 is 0.843. SVR-front4 reached the highest FM when c=0.2, as shown in Figure 27 (a). The highest FM for SVR-front2 is 0.933, which can be reached when c=0.13. The corresponding test result is shown in Figure 27 (b). The FM for SVC-front4 and SVC-front2 are 0.407 and 0.447 respectively.

Table 2: Comparison of classification models using classification accuracy, and the comparisons of models using modified classification accuracy

	CA	Fp1, Fp2, F3, F4 (SVC)	Fp1, Fp2 (SVC)	MCA	Fp1, Fp2, F3, F4 (SVC)	Fp1, Fp2 (SVC)
Fp1, Fp2 (SVC)	0.858	p = 0.858		0.957	p = 0.742	
Fp1, Fp2 (LGS)	0.755	p < 0.001	p < 0.001	0.893	p < 0.001	p < 0.001
Fp1, Fp2, T3, T4 (SVC)	0.596	p < 0.005	p < 0.005	0.754	p < 0.001	p < 0.001
Fp1, Fp2, T3, T4 (LGS)	0.665	p < 0.001	p < 0.001	0.837	p < 0.001	p < 0.001
Fp1, Fp2, F3, F4 (SVC)	**0.860**			**0.960**		
Fp1, Fp2, F3, F4 (LGS)	0.752	p < 0.001	p < 0.005	0.884	p < 0.001	p < 0.005
T3, T4, F3, F4 (SVC)	0.583	p < 0.001	p < 0.005	0.729	p < 0.001	p < 0.001
T3, T4, F3, F4 (LGS)	0.646	p < 0.001	p < 0.001	0.831	p < 0.001	p < 0.001
Fp1, Fp2, T3, T4, F3, F4 (SVC)	0.597	p < 0.005	p < 0.005	0.765	p < 0.001	p < 0.005
Fp1, Fp2, T3, T4, F3, F4 (LGS)	0.681	p < 0.001	p < 0.001	0.846	p < 0.001	p < 0.001

Table 3: Comparison of regression models using balanced mean absolute error

	MAE	Fp1, Fp2, F3, F4 (SVR)	Fp1, Fp2 (SVR)
Fp1, Fp2 (RDG)	0.467	p < 0.001	p < 0.001
Fp1, Fp2 (LSS)	0.471	p < 0.001	p < 0.001
Fp1, Fp2 (SVR)	0.383	p = 0.215	
Fp1, Fp2, T3, T4 (RDG)	0.458	p < 0.001	p < 0.001
Fp1, Fp2, T3, T4 (LSS)	0.480	p < 0.001	p < 0.001
Fp1, Fp2, T3, T4 (SVR)	0.480	p < 0.05	p = 0.069
Fp1, Fp2, F3, F4 (RDG)	0.453	p < 0.005	p < 0.005
Fp1, Fp2, F3, F4 (LSS)	0.459	p < 0.001	p < 0.001
Fp1, Fp2, F3, F4 (SVR)	**0.372**		
T3, T4, F3, F4 (RDG)	0.475	p < 0.001	p < 0.001
T3, T4, F3, F4 (LSS)	0.505	p < 0.001	p < 0.001
T3, T4, F3, F4 (SVR)	0.469	p < 0.01	p < 0.05
Fp1, Fp2, T3, T4, F3, F4 (RDG)	0.456	p < 0.001	p < 0.001
Fp1, Fp2, T3, T4, F3, F4 (LSS)	0.488	p < 0.001	p < 0.001
Fp1, Fp2, T3, T4, F3, F4 (SVR)	0.436	p < 0.01	p < 0.05

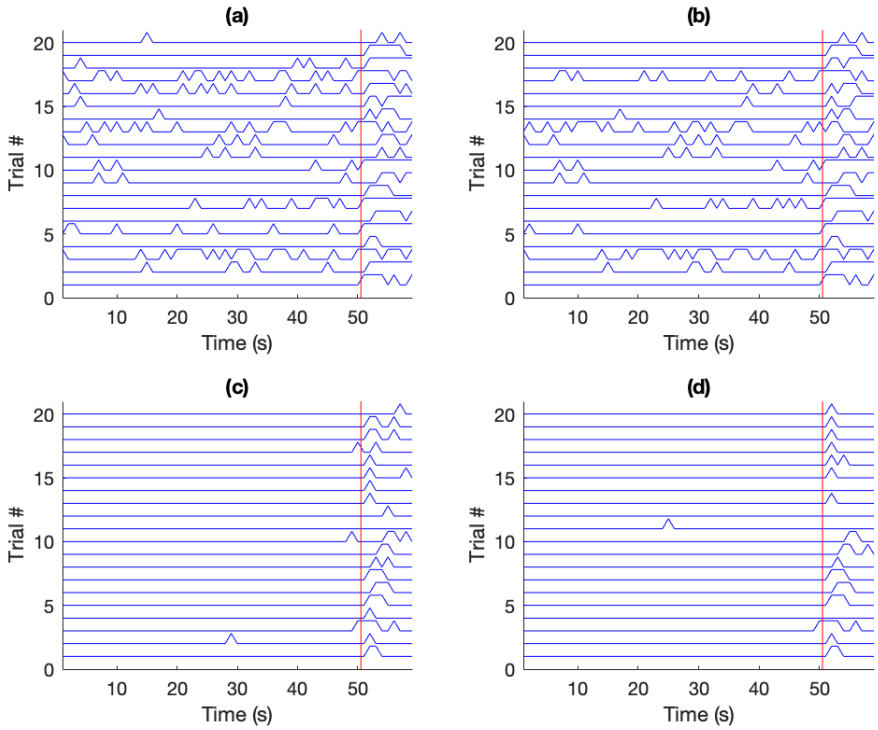


Figure 26: Test dataset prediction results of (a) SVC-front4, (b) SVC-front2, (c) SVR-front4 ($c = 0.2$), and (d) SVR-front2 ($c = 0.2$). The red lines are stage changing times. The peaks are positive fear response predictions.

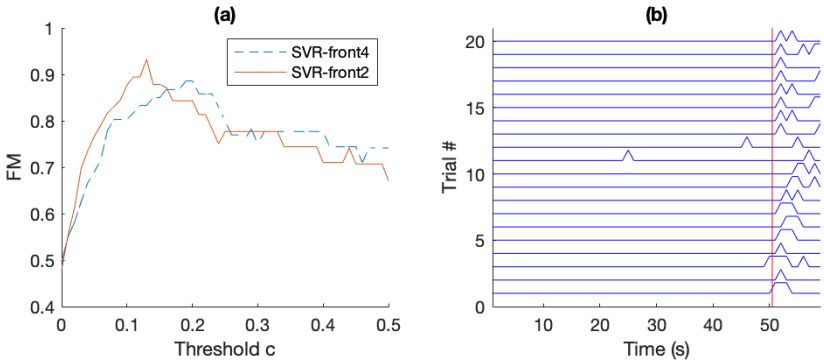


Figure 27: (a) FM of SVR-front4 and SVR-front2 under various thresholds and (b) Prediction results of SVR-front2 $c = 0.13$.

5.10 CONCLUSSION AND DISCUSSIONS

We designed an experiment to collect the EEG when the human subjects feel dangerous while working collaboratively with an industrial robot. The fear emotion generated in our experiment is different from the previous experiments. In the previous experiment, the fear signal is collected between the subjects received a hint and received an electrical shock. In other word, the subjects generated fear emotion while waiting for the electrical shock. However, in our experiment, the fear emotion is generated while watching the unexpected motions of the robot. Rather than being afraid, there are some worries and frights also. So, we think the emotions in the EEG are not only fear itself in our experiment. The unsafe feeling of the human while the robot is making un-predictable motions is not pure fear. Thus, the "fear" emotion detected in our study is only for human-robot collaborative working.

The fear response did not stay in the same level during the experiment stage 2. As shown in Figure 27, the averaged PSD suggested that the emotion reached the peak at the *3*-rd second after the experiment stage change, then decayed in the rest of the experiment. The delay of the peak can be related to the robot trajectory in the experiment. That is, in the beginning of the stage 2, the motion of the robot was not being afraid by the emotion of the subjects did not contain much fear. For classification models, it is not trivial to manually distinguish which second the subjects feel dangerous, which second don't. But, if all the EEG segments were labeled as fear response (the same as we did in this study), there results a lot of false positives in the stage 1. On the other hand, regression models handled this situation much better as we demonstrated in the study.

For regression models, we applied the equation (3) to determine which prediction is a fear response. The equation (3) basically classify a prediction as a fear response if it is higher than the historical mean for two standard deviations. However, the standard deviation can be very small when we only have a few historical data. So, a threshold was added in the equation (3) to control the detection sensitivity.

CHAPTER 6 : FUTURE WORK

In this work, we developed BCIs to interact with industrial robots actively or passively. In the active interaction, the operator can send commands to the robot with SSVEP. In the passive interaction, the robot will observe the fear response from the operator's EEG. We also introduced a CNN and CCA combined model "Conv-CA" which works great on the 40-frequencies SSVEP classification task. We think there are some future work directions that are worth well to explore:

1) For the active interactions, the SSVEP we are currently using is a signal that is very stable and easy to detect. However, the signal has some major drawbacks. First, the command sender has to stare at a flashing light. This action is not intuitive for giving a command. Second, if the flashing light is not strong enough or the flashing image is not big enough, SSVEP can be difficult to detect. Third, SSVEP doesn't work for everyone. Some people can have very weak SSVEP. Ideally, what we want to send is our thought or even a simple 'yes' or 'no'. But that is almost impossible to be extracted in EEG. And, when people try hard to think, some EMG signal can mess up EEG. We think, if a binary thought like "yes" and "no" can be detected in EEG, or people can learn a way to think "yes" and "no" such that the thought can be captured in EEG, combining with eye gazing technology, it will be a great human-robot interaction.
2) Comparing to active interaction, we think there are more possibilities in passive interaction. The passive interaction is intuitive, because people can just focus on their job while the BCI functions by itself. It can be used to detect certain response, for example the unsafe

feeling we studied, or some emotions, like happy, depressed, sleepy, boring, etc. These emotions detection can benefit social robots, safety driving, rehabilitation training, etc. The current research collects EEG with emotions by showing emotion relevant videos or images to experiment subjects. But the classification accuracy is pretty low. We think alternative experiment designs should be considered in order to collect EEG with higher quality emotions. The experiment must be designed carefully so that the subjects can achieve target emotions in the experiment. And we want the experiment can be repeated on the same subject for multiple times.

3) As for the classification model of noisy biomedical signals, deep learning models can be a strong tool. As we discussed in the chapter 4, complicated deep learning models can result strong overfitting on low SNR signals. Combining adding a deterministic layer in the deep learning model, for example the CCA layer in Conv-CA, can prevent overfitting.

REFERENCES

[1] Niedermeyer, Ernst, and FH Lopes da Silva, eds. *Electroencephalography: basic principles, clinical applications, and related fields*. Lippincott Williams & Wilkins, 2005.

[2] Bozinovski, Stevo, Mihail Sestakov, and Liljana Bozinovska. "Using EEG alpha rhythm to control a mobile robot." In *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pp. 1515-1516. IEEE, 1988.

[3] Haddadin, Sami, Alin Albu-Schaffer, Alessandro De Luca, and Gerd Hirzinger. "Collision detection and reaction: A contribution to safe physical human-robot interaction." In *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3356-3363. IEEE, 2008.

[4] Hirzinger, Gerd, Norbert Sporer, Alin Albu-Schaffer, M. Hahnle, Rainer Krenn, Antonio Pascucci, and Markus Schedl. "DLR's torque-controlled light weight robot III-are we reaching the technological limits now?." In *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No. 02CH37292)*, vol. 2, pp. 1710-1716. IEEE, 2002.

[5] Zinn, Michael, Bernard Roth, Oussama Khatib, and J. Kenneth Salisbury. "A new actuation approach for human friendly robot design." *The international journal of robotics research* 23, no. 4-5 (2004): 379-398.

[6] Shin, Dongjun, Akichika Tanaka, Namho Kim, and Oussama Khatib. "A centrifugal force-based configuration-independent high-torque-density passive brake for human-friendly robots." *IEEE/ASME Transactions on Mechatronics* 21, no. 6 (2016): 2827-2835.

[7] Geravand, Milad, Fabrizio Flacco, and Alessandro De Luca. "Human-robot physical interaction and collaboration using an industrial robot with a closed control architecture." In *2013 IEEE International Conference on Robotics and Automation*, pp. 4000-4007. IEEE, 2013.

[8] Zanchettin, Andrea Maria, Nicola Maria Ceriani, Paolo Rocco, Hao Ding, and Björn Matthias. "Safety in human-robot collaborative manufacturing environments: Metrics and control." *IEEE Transactions on Automation Science and Engineering* 13, no. 2 (2015): 882-893.

[9] Wilcox, Ronald, Stefanos Nikolaidis, and Julie Shah. "Optimization of temporal dynamics for adaptive human-robot interaction in assembly manufacturing." *Robotics* 8 (2013): 441.

[10] Haddadin, Sami, Alin Albu-Schäffer, and Gerd Hirzinger. "Requirements for safe robots: Measurements, analysis and new insights." *The International Journal of Robotics Research*28, no. 11-12 (2009): 1507-1527.

[11] Cordero, Cristina Alén, Giuseppe Carbone, Marco Ceccarelli, Javier Echávarri, and José Luis Muñoz. "Experimental tests in human–robot collision evaluation and characterization of a new safety index for robot operation." *Mechanism and machine theory* 80 (2014): 184-199.

[12] Haddadin, Sami, Alin Albu-Schaffer, and Gerd Hirzinger. "The role of the robot mass and velocity in physical human-robot interaction-Part I: Non-constrained blunt impacts." In *2008 IEEE International Conference on Robotics and Automation*, pp. 1331-1338. IEEE, 2008.

[13] Gemignani, Guglielmo, Manuela Veloso, and Daniele Nardi. "Language-based sensing descriptors for robot object grounding." In *Robot Soccer World Cup*, pp. 3-15. Springer, Cham, 2015.

[14] Sheikholeslami, Sara, AJung Moon, and Elizabeth A. Croft. "Exploring the effect of robot hand configurations in directional gestures for human-robot interaction." In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3594-3599. IEEE, 2015.

[15] Gleeson, Brian, Karon MacLean, Amir Haddadi, Elizabeth Croft, and Javier Alcazar. "Gestures for industry Intuitive human-robot communication from human observation." In *2013 8th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pp. 349-356. IEEE, 2013.

[16] Beetz, Michael, Georg Bartels, Alin Albu-Schäffer, Ferenc Bálint-Benczédi, Rico Belder, Daniel Beßler, Sami Haddadin et al. "Robotic agents capable of natural and safe physical interaction with human co-workers." In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 6528-6535. IEEE, 2015.

[17] Oshin, Olusegun, Edgar A. Bernal, Binu M. Nair, Jerry Ding, Richa Varma, Richard W. Osborne, Eddie Tunstel, and Francesca Stramandinoli. "Coupling Deep Discriminative and Generative Models for Reactive Robot Planning in Human-Robot Collaboration." In *2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)*, pp. 1869-1874. IEEE, 2019.

[18] Vogel, Jörn, Sami Haddadin, John D. Simeral, Sergey D. Stavisky, Daniel Bacher, Leigh R. Hochberg, John P. Donoghue, and Patrick Van Der Smagt. "Continuous control of the dlr light-weight robot iii by a human with tetraplegia using the braingate2 neural interface system." In *Experimental Robotics*, pp. 125-136. Springer, Berlin, Heidelberg, 2014.

[19] Edlinger, Günter, Clemens Holzner, and Christoph Guger. "A hybrid brain-computer interface for smart home control." In *International conference on human-computer interaction*, pp. 417-426. Springer, Berlin, Heidelberg, 2011.

[20] Riaz, Anaum, Sana Akhtar, Shanza Iftikhar, Amir Ali Khan, and Ahmad Salman. "Inter comparison of classification techniques for vowel speech imagery using EEG sensors." In *The 2014 2nd International Conference on Systems and Informatics (ICSAI 2014)*, pp. 712-717. IEEE, 2014.

[21] Yin, Erwei, Zongtan Zhou, Jun Jiang, Fanglin Chen, Yadong Liu, and Dewen Hu. "A novel hybrid BCI speller based on the incorporation of SSVEP into the P300 paradigm." *Journal of neural engineering* 10, no. 2 (2013): 026012.

[22] Ying, Robert, Jonathan Weisz, and Peter K. Allen. "Grasping with your brain: a brain-computer interface for fast grasp selection." In *Robotics Research*, pp. 325-340. Springer, Cham, 2018.

[23] Hortal, Enrique, Daniel Planelles, A. Costa, Eduardo Iánez, Andrés Úbeda, José Maria Azorín, and Eduardo Fernández. "SVM-based Brain–Machine Interface for controlling a robot arm through four mental tasks." *Neurocomputing* 151 (2015): 116-121.

[24] Gandhi, Vaibhav, Girijesh Prasad, Damien Coyle, Laxmidhar Behera, and Thomas Martin McGinnity. "EEG-based mobile robot control through an adaptive brain–robot interface." *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 44, no. 9 (2014): 1278-1285.

[25] LaFleur, Karl, Kaitlin Cassady, Alexander Doud, Kaleb Shades, Eitan Rogin, and Bin He. "Quadcopter control in three-dimensional space using a noninvasive motor imagery-based brain–computer interface." *Journal of neural engineering* 10, no. 4 (2013): 046003.

[26] Chen, Liang-Chieh, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. "Encoder-decoder with atrous separable convolution for semantic image segmentation." In *Proceedings of the European conference on computer vision (ECCV)*, pp. 801-818. 2018.

[27] Dong, Jianwei, Yanmei Chen, Zhijing Yang, and Bingo Wing-Kuen Ling. "A parallel thinning algorithm based on stroke continuity detection." *Signal, Image and Video Processing* 11, no. 5 (2017): 873-879.

[28] Zhang, Y. U., Guoxu Zhou, Jing Jin, Xingyu Wang, and Andrzej Cichocki. "Frequency recognition in SSVEP-based BCI using multiset canonical correlation analysis." *International journal of neural systems* 24, no. 04 (2014): 1450013.

[29] Lin, Zhonglin, Changshui Zhang, Wei Wu, and Xiaorong Gao. "Frequency recognition based on canonical correlation analysis for SSVEP-based BCIs." *IEEE transactions on biomedical engineering* 53, no. 12 (2006): 2610-2614.

[30] Bin, Guangyu, Xiaorong Gao, Yijun Wang, Yun Li, Bo Hong, and Shangkai Gao. "A high-speed BCI based on code modulation VEP." *Journal of neural engineering* 8, no. 2 (2011): 025015.

[31] Resalat, Seyed Navid, and Seyed Kamaledin Setarehdan. "An improved ssvep based bci system using frequency domain feature classification." *American Journal of Biomedical Engineering* 3, no. 1 (2013): 1-8.

[32] Peng, Nengneng, et al. "Control of a nursing bed based on a hybrid brain-computer interface." *2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE, 2016.

[33] Lo, Chi-Chun, Shang-Ho Tsai, and Bor-Shyh Lin. "Novel non-contact control system of electric bed for medical healthcare." *Medical & biological engineering & computing* 55.3 (2017): 517-526.

[34] Li, Yao, and Thenkurussi Kesavadas. "Brain computer interface robotic co-workers: Defective part picking system." *ASME 2018 13th International Manufacturing Science and Engineering Conference*. American Society of Mechanical Engineers Digital Collection, 2018.

[35] Li, Yao, and Thenkurussi Kesavadas. "Welding Robotic Co-Worker Using Brain Computer Interface." *ASME 2018 International Mechanical Engineering Congress and Exposition*. American Society of Mechanical Engineers Digital Collection, 2018.

[36] Nakanishi, Masaki, et al. "Enhancing detection of SSVEPs for a high-speed brain speller using task-related component analysis." *IEEE Transactions on Biomedical Engineering* 65.1 (2017): 104-112.

[37] Nguyen, Trung-Hau, Da-Lin Yang, and Wan-Young Chung. "A high-rate BCI speller based on eye-closed EEG signal." *IEEE Access* 6 (2018): 33995-34003.

[38] Chai, Xiaoke, et al. "A hybrid BCI-controlled smart home system combining SSVEP and EMG for individuals with paralysis." *Biomedical Signal Processing and Control* 56 (2020): 101687.

[39] Saboor, Abdul, et al. "SSVEP-based BCI in a smart home scenario." *International Work-Conference on Artificial Neural Networks*. Springer, Cham, 2017.

[40] Cruz, Inês, et al. "Kessel Run-A Cooperative Multiplayer SSVEP BCI Game." *International Conference on Intelligent Technologies for Interactive Entertainment*. Springer, Cham, 2017.

[41] Chen, Shih-Chung, et al. "A Single-Channel SSVEP-Based BCI with a Fuzzy Feature Threshold Algorithm in a Maze Game." *International Journal of Fuzzy Systems* 19.2 (2017): 553-565.

[42] Koo, Bonkon, et al. "Immersive BCI with SSVEP in VR head-mounted display." *2015 37th annual international conference of the IEEE engineering in medicine and biology society (EMBC)*. IEEE, 2015.

[43] Chen, Xiaogang, et al. "Control of a 7-DOF robotic arm system with an SSVEP-based BCI." *International journal of neural systems* 28.08 (2018): 1850018.

[44] Chiuzbaian, Andrei, Jakob Jakobsen, and Sadasivan Puthusserypady. "Mind Controlled Drone: An Innovative Multiclass SSVEP based Brain Computer Interface." *2019 7th International Winter Conference on Brain-Computer Interface (BCI)*. IEEE, 2019.

[45] Waytowich, Nicholas R., and Dean J. Krusienski. "Development of an extensible SSVEP-BCI software platform and application to wheelchair control." *2017 8th International IEEE/EMBS Conference on Neural Engineering (NER)*. IEEE, 2017.

[46] Chen, Xiaogang, et al. "Combination of high-frequency SSVEP-based BCI and computer vision for controlling a robotic arm." *Journal of neural engineering* (2018).

[47] Stawicki, Piotr, Felix Gembler, and Ivan Volosyak. "Driving a semiautonomous mobile robotic car controlled by an SSVEP-based BCI." *Computational intelligence and neuroscience*2016 (2016): 5.

[48] Regan, David. "Human brain electrophysiology." *Evoked potentials and evoked magnetic fields in science and medicine* (1989).

[49] Friman, Ola, Ivan Volosyak, and Axel Gräser. "Multiple channel detection of steady-state visual evoked potentials for brain-computer interfaces." *Biomedical Engineering, IEEE Transactions on* 54.4 (2007): 742-750.

[50] Y. Zhang et al., "Multiway canonical correlation analysis for frequency components recognition in SSVEP-based BCIs," in Proc. 18th Int. Conf. Neural Inform. Process., 2011, pp. 287–295.

[51] Y. Zhang et al., "L1-regularized multiway canonical correlation analysis for SSVEP-based BCI," IEEE Trans. Neural Syst. Rehabil. Eng., vol. 21, no. 6, pp. 887–896, Nov. 2013.

[52] Y. Zhang et al., "Frequency recognition in SSVEP-based BCI using multi- set canonical correlation analysis," Int. J. Neural Syst., vol. 24, no. 4, 2014, Art. no. 1450013.

[53] Chen, Xiaogang, et al. "Filter bank canonical correlation analysis for implementing a high-speed SSVEP-based brain–computer interface." Journal of neural engineering 12.4 (2015): 046008.

[54] Y. Wang et al., “Enhancing detection of steady-state visual evoked potentials using individual training data,” in Proc. 36th Ann. Int. Conf. IEEE Eng. Med. Biol. Soc., 2014, pp. 3037–3040.

[55] Nakanishi, Masaki, et al. "Enhancing detection of SSVEPs for a high-speed brain speller using task-related component analysis." IEEE Transactions on Biomedical Engineering 65.1 (2017): 104-112.

[56] Zhang, Yangsong, et al. "Correlated component analysis for enhancing the performance of SSVEP-based brain-computer interface." *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 26.5 (2018): 948-956.

[57] Aznan, Nik Khadijah Nik, et al. "On the classification of SSVEP-based dry-EEG signals via convolutional neural networks." *2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE, 2018.

[58] Kwak, No-Sang, Klaus-Robert Müller, and Seong-Whan Lee. "A convolutional neural network for steady state visual evoked potential classification under ambulatory environment." *PloS one* 12.2 (2017): e0172578.

[59] Podmore, Joshua J., et al. "On the relative contribution of deep convolutional neural networks for SSVEP-based bio-signal decoding in BCI speller applications." *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 27.4 (2019): 611-618.

[60] Wang, Yijun, et al. "A benchmark dataset for SSVEP-based brain–computer interfaces." *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 25.10 (2016): 1746-1752.

[61] Chen, Chaona, Oliver GB Garrod, Jiayu Zhan, Jonas Beskow, Philippe G. Schyns, and Rachael E. Jack. "Reverse engineering psychologically valid facial expressions of emotion into social robots." In *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)*, pp. 448-452. IEEE, 2018.

[62] Churamani, Nikhil, Matthias Kerzel, Erik Strahl, Pablo Barros, and Stefan Wermter. "Teaching emotion expressions to a human companion robot using deep neural architectures." In *2017 International Joint Conference on Neural Networks (IJCNN)*, pp. 627-634. IEEE, 2017.

[63] Cominelli, Lorenzo, Nicola Carbonaro, Daniele Mazzei, Roberto Garofalo, Alessandro Tognetti, and Danilo De Rossi. "A multimodal perception framework for users emotional state assessment in social robotics." *Future Internet* 9, no. 3 (2017): 42.

[64] Tsiourti, Christiana, Astrid Weiss, Katarzyna Wac, and Markus Vincze. "Multimodal integration of emotional signals from voice, body, and context: Effects of (in) congruence on emotion recognition and attitudes towards robots." *International Journal of Social Robotics* 11, no. 4 (2019): 555-573.

[65] Williams, W. Craig, Yuan Chang Leong, and Eleanor A. Collier. "Neural correlates of communicating emotion through facial expressions: A simultaneous fMRI and facial EMG study." (2019).

[66] Katsigiannis, Stamos, and Naeem Ramzan. "DREAMER: A database for emotion recognition through EEG and ECG signals from wireless low-cost off-the-shelf devices." IEEE journal of biomedical and health informatics 22, no. 1 (2017): 98-107.

[67] Song, Tengfei, Wenming Zheng, Peng Song, and Zhen Cui. "EEG emotion recognition using dynamical graph convolutional neural networks." IEEE Transactions on Affective Computing (2018).

[68] Zheng, Wei-Long, Jia-Yi Zhu, and Bao-Liang Lu. "Identifying stable patterns over time for emotion recognition from EEG." IEEE Transactions on Affective Computing (2017).

[69] Koelstra, Sander, Christian Muhl, Mohammad Soleymani, Jong-Seok Lee, Ashkan Yazdani, Touradj Ebrahimi, Thierry Pun, Anton Nijholt, and Ioannis Patras. "Deap: A database for emotion analysis; using physiological signals." IEEE transactions on affective computing 3, no. 1 (2011): 18-31.

[70] Johnson, Michael R., and Lawrence E. Adler. "Transient impairment in P50 auditory sensory gating induced by a cold-pressor test." *Biological psychiatry* 33, no. 5 (1993): 380-387.

[71] White, Patricia M., and Cindy M. Yee. "Effects of attentional and stressor manipulations on the P50 gating response." *Psychophysiology* 34, no. 6 (1997): 703-711.

[72] LaBar, Kevin S., J. Christopher Gatenby, John C. Gore, Joseph E. LeDoux, and Elizabeth A. Phelps. "Human amygdala activation during conditioned fear acquisition and extinction: a mixed-trial fMRI study." *Neuron* 20, no. 5 (1998): 937-945.

[73] Fullana, M. A., B. J. Harrison, C. Soriano-Mas, Bram Vervliet, N. Cardoner, A. Àvila-Parcet, and J. Radua. "Neural signatures of human fear conditioning: an updated and extended meta-analysis of fMRI studies." *Molecular Psychiatry*21, no. 4 (2016): 500-508.

[74] Armony, Jorge L., and Raymond J. Dolan. "Modulation of spatial attention by fear-conditioned stimuli: an event-related fMRI study." *Neuropsychologia* 40, no. 7 (2002): 817-826.

[75] Kisley, Michael A., and Zoe M. Cornwell. "Gamma and beta neural activity evoked during a sensory gating paradigm: effects of auditory, somatosensory and cross-modal stimulation." *Clinical neurophysiology* 117, no. 11 (2006): 2549-2563.