Biomedical Signal Processing and Control 84 (2023) 104765

Contents lists available at ScienceDirect

Biomedical Signal Processing and Control

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

Autonomous grasping of 3-D objects by a vision-actuated robot arm using Brain–Computer Interface

Arnab Rakshit [a,*], Shraman Pramanick [b,1], Anurag Bagchi [a,1], Saugat Bhattacharyya [c]

[a] *Department of Electronics & Tele-communication Engineering, Jadavpur University, India*
[b] *Department of Electrical & Computer Engineering, Johns Hopkins University, USA*
[c] *School of Computing, Engineering and Intelligent Systems, Ulster University, United Kingdom*

ARTICLE INFO

Keywords:
Brain–Computer Interface
Electroencephalogram
Robot
Convolutional Neural Network
Overlapping Object GraspNet

ABSTRACT

A major drawback of a Brain–Computer Interface-based robotic manipulation is the complex trajectory planning of the robot arm to be carried out by the user for reaching and grasping an object. The present paper proposes an intelligent solution to the existing problem by incorporating a novel Convolutional Neural Network (CNN)-based grasp detection network that enables the robot to reach and grasp the desired object (including overlapping objects) autonomously using a RGB-D camera. This network uses a simultaneous object and grasp detection to affiliate each estimated grasp with its corresponding object. The subject uses motor imagery brain signals to control the pan and tilt angle of a RGB-D camera mounted on a robot link to bring the desired object inside its Field-of-view presented through a display screen while the objects appearing on the screen are selected using the P300 brain pattern. The robot uses inverse kinematics along with the RGB-D camera information to autonomously reach the selected object and the object is grasped using proposed grasping strategy. The overall BCI system outperforms other comparative systems involving manual trajectory planning significantly. The overall accuracy, steady-state error, and settling time of the proposed system are 93.4%, 0.05%, and 15.92 s, respectively. The system also shows a significant reduction of the workload of the operating subjects in comparison to manual trajectory planning based approaches for reaching and grasping.

1. Introduction

People suffering from neuro-motor disabilities face great difficulty in locating and grasping objects even if the desired object is present within their reach. With the recent development of Brain–Computer Interface (BCI) technology and current state-of-the-art robotic arms, hands and perception systems, it has been proved that these individuals with restricted mobility can interact with their environment to perform activities of daily living (ADL), including things like drinking water, opening doors, and other basic actions. BCI provides a direct non muscular communication between the neural activity generated by the subject's brain and the outside world [1]. For electroencephalography (EEG) based non-invasive BCI, the brain signals are obtained by placing the electrodes on the surface of subject's scalp which are then mapped to manipulate external devices such as humanoid robots [2,3], virtual helicopters [4,5], wheelchairs [6,7], tele-presence mobile robots [8,9]. In the recent past, BCI has been successfully used for rehabilitation training of stroke patients [10–12], motor control of prosthetic limbs [13–15] and performing several activities of daily living [16–18]. However, accurate object grasping using only brain-commanded signals is still an open challenge because of the high degrees of freedom (DOF) and challenges arising from complex precise position control of the robot arm.

EEG based BCI can be categorized based on characteristic brain activity patterns. Among them Motor Imagery (Event-Related Desynchronization/Synchronization (ERD/ERS)) [19,20], Steady State Visual Evoke Potentials (SSVEP) [21,22] and P300 patterns [23] are widely used. Motor Imagery is extensively used for control of brain-actuated robot link control and navigation. However the main drawback of the MI based system is the rigorous subject training required. P300-based BCI is relatively easy to use for generating control signals without extensive training of the user. There are also traces of work where a hybrid modality employing two or more brain signals is used for robot link manipulation. But most of the previous research works employing MI and P300 are solely based on subjective control of the robot link

[*] Corresponding author.
E-mail addresses: arnabrakshit2008@gmail.com (A. Rakshit), shraman.pramanick@gmail.com (S. Pramanick), miccooper9@gmail.com (A. Bagchi), saugatbhattacharyya@gmail.com (S. Bhattacharyya).
[1] Equal Contribution.

https://doi.org/10.1016/j.bspc.2023.104765
Received 6 May 2022; Received in revised form 5 February 2023; Accepted 22 February 2023
Available online 7 March 2023

where the participating subject mentally guides the robot arm to reach and grasp the desired object. The main drawbacks of those systems are two folds, first, the brain commanded robot often misses the target object resulting a large positive or negative positional error. Second, it requires large amount of subject training to accurately control the position of robot arm to grasp the desired object. Here the subjects need to perform the complex trajectory planning of the robot arm in order to align the robot gripper with desired object. It becomes extremely difficult for the human subject to perform such complex planning mentally and to control the position and orientation of the robot gripper to perfectly grip the desired object. Added to the above facts, such complex trajectory planning imposes a high cognitive load in the user's brain.

Literature shows that quite a few studies exist in the domain of EEG based grasping control. In 2012, Hochberg et al. [24] and Collinger et al. [25] developed neural interface system-based control of robotic arms to perform three-dimensional reach and grasp movements for patients with tetraplegia. Later in the past few years ample experiment is conducted to control the robot arm using brain signals for reaching and grasping task [26–28] In all the above cases, MI based BCI protocol is used. These BCI systems required several weeks of training sessions to learn the direct motor control with high DOF. Such a rigorous training procedure often causes a large discomfort to the participants. Moreover, as these systems are not fully autonomous and the user controls the complete trajectory of the robot-arm, their performance is limited for real-time practical applications and the cognitive load of the operating subjects are also increased. There also exists quite a few literature which uses P300 navigational signal to the robot. Spataro et al. [29] used P300 based EEG command to control a humanoid robot with the aim of reaching and grasping a glass of water; however the desired object cannot be directly selected and the subject needs to mentally guide (using P300 based GUI) the manipulator to reach the object. Such drawback is also seen in few other works as well [30–32] As discussed, such strategy imposes a high cognitive load in the subjects' brain. Recently, Rakshit et al. [33] proposed a SSVEP based random order robot link selection and P300 based link movement seizing strategy to reduce the positional error. They found a drastic reduction in positional error compared to the other state of the art literature [34], but here again, the entire robot arm trajectory is planned by the subject.

BCI based shared control strategy has also been studied in the past. Tang et al. [35] used a shared control strategy to grasp an object using robot hand but the method suffers from the fact that the objects used for the experiment were identical in nature (red cap bottles), which makes the method challenging for diverse objects present in the environment. Xu et al. [36] proposed a novel shared control strategy where subjects mentally guided a robot end-effector in a horizontal plane and once the end-effector comes within close vicinity of the target object, the switch over to automatic control using vision-based movement-planning is instigated. Although they achieved the highest accuracy around 97% but the following points still need to be addressed there. First, the user can move the end-effector in only horizontal plane, no control commands are given to move it in vertical direction. Second, the scheme does not allow the user to select the target object priory, he/she still has to mentally plan the trajectory of the end-effector and use motor imagery to reach the target object. In their continuation work, Xu et al. [37]extended their strategy for multiple objects and provided adaptive assistance to the participating subject. Assistance was provided by implementing autonomous trajectory correction and autonomous grasping during reaching and grasping of the target object respectively. The scheme still requires human intervention in the path planning of the end-effector. Grasping performance was evaluated using three identical objects scattered in workspace, hence its performance for various objects in various scene (overlapping and non-overlapping) is still to be explored. In [38] Liu et al. proposed a novel strategy of controlling a dual arm robot using motor imagery and a kinect sensor. The subject used their left and right MI to command a dual arm robot to lift and drop a given object respectively. A PDNN based neuro-dynamics optimization was used for solving the motion redundancy of the robotic arm. In [39] Tang et al. proposed an BCI based robot manipulation approach to quickly grasp a object using motor imagery and camera based object detection technology. A camera is used here to capture the live feed of the robot environment which is visible to the user through a computer monitor. The subject observes the computer screen and uses left/right arm motor imagery to align the robot arm in such a way that the target object should come in the target area(center of the camera view). The YOLO object detection algorithm is used to get the information about the object inside the target area and the grasp command is executed thereafter. We recognize first that aligning the target object with target using mental commands is bit challenging for the patients and second the performance of the system is still unknown for objects located spatially very close to each other (for the condition when more than one object come inside the target area). Recently Zeng et al. [40]proposed a novel shared controller which dynamically blends the user motion planning and autonomous motion planning to achieve a smooth and collision free robot trajectory. The user continuously uses his/her gaze direction to move the end-effector in a desired direction over a horizontal plane and simultaneously performs motor imagination to modulate the speed of it. The subject gets assistance for most difficult part of the task. The strategy yields a maximum of 100% success rate in this context. However the paper focuses mostly on the reaching task and its performance(reaching+grasping) in presence of multiple overlapping objects is yet to be explored. The scheme also involves user intervention throughout the task (focusing gaze and performing MI simultaneously), which may increase the cognitive load of the novel participating subjects. Duan et al. [2] proposed an approach to manipulate wheeled robot using mental commands and computer vision. A camera mounted on the chest of the mobile robot extracts the information about the robot environment. The computer screen displaying camera-view about robot's trajectory of motion includes provisions for generating navigational commands for the robot using SSVEP, whereas MI commands are issued to accomplish the manipulation task such as grasping the object. The grasping phase was validated using a object which carries a color mark on its body which helps the vision system to distinguish the object from the background based on the color feature. Hence the proposed system performance is yet to be explored in real life scenario, where multiple objects with different colors are present and it is also difficult to mark each of them with color marker. Wang et al. [41] in a recent work used camera based real time feedback to navigate a tele-presence robot and reach the desired object using SSVEP. The scheme employs a camera mounted on top of a robot arm to explore the objects within the field of view, which are transferred to a computer monitor for selecting the target object using SSVEP. Here for each object, a bounding box is developed. These bounding boxes flicker at different frequencies to represent the identity of the individual objects. A subject intending of selecting a specific object focuses on the item and the flickering frequency is picked up by the subject through SSVEP. Once the object is selected, the navigation of the robot arm is automated by camera-based position control system. However if the objects are located spatially very close to each other, it might fall inside the same flickering bounding box making it difficult to grasp any one of them. Apart from that, prolonged attention over the SSVEP stimuli also causes mental and eye fatigue to the user. Similarly, Zhang et al. in [27] used a MI based shared control strategy to control the robot grasping and in their another paper [42] they used SSVEP based object selection along with the Kinect based machine vision, the shared control strategy is used for the same purpose. Recently, Di Lillo P et al. [43] used a similar P300 and Kinect based grasping strategy to control a manipulator in grasping an object. Li et al. developed a BCI based shared control strategy to navigate a humanoid robot by combining central vision tracking strategy and two different brain signals N200 and P300 [44]. Batzianouliset al. [45] proposed an interesting approach for shared control by utilizing ErrP signal and Inverse reinforcement learning (IRL) paradigm. Here the subject has its own choice of trajectory planning of a robot, for instance obstacle avoidance and trajectory planning towards fixed target. Each time the robot reaches an obstacle before reaching the destination, the subject releases an ErrP signal. The decoded ErrP signal is used to move away the robot from the obstacle without violating the planned trajectory of the robot. An IRL algorithm is employed to change the trajectory following user's preferred trajectory of motion.

In all the literature stated above, the authors have relied on existing object recognition strategy which is not capable of detecting multiple overlapping objects present in the field of view of camera. In real life scenario such multiple overlapping objects can frequently be found in domestic workspace of the user and grasping of those objects using only manual cognitive effort is immensely difficult for a human user.

In this paper, we aim to provide an intelligent solution to this problem of brain-actuated object grasping with the help of a camera mounted on a robot arm to localize the object on the computer monitor and autonomously control the motion of the arm to accurately grasp the object. The paper employs a 6-DOF robot arm and a Microsoft Kinect which can be used as a depth sensing device in association with an RGB camera. Surrounding environment is visible to the subject in the computer screen through the real time feed of the kinect. The kinect is mounted on the robot arm so that it can move in accordance with the arm. The subject uses feet motor imagery to choose between pan and tilt motion of the camera and hand motor imagery to change the pan and tilt angle of the camera(by moving the 1st/5th joint of the robot arm) until the desired object comes into the field of view of kinect and appears in the screen. A standard pre-trained 1-D CNN classifier is used to decode the hand and feet motor imagery signal. A deep learning based masking algorithm is used to estimate the object masks in the environment and to compute the centroid of all the objects appearing in the screen. To choose the desired object, the centroid of the objects appearing in the screen are flashed randomly. Once the centroid of target object is flashed, the subject releases a P300 signal, indicating his/her choice about the desired object. After the desired object is known to the system, the robot arm automatically moves closer to the object and we employ a novel Overlapping Object Grasping Network(OOGNet) to estimate proper grasping rectangle and finally, grip the desired object by a parallel-plate gripper mounted at the end of the last link of the arm.

Our main contributions are summarized as follows

- We present a novel brain-commanded object grasping scheme to localize, select and grasp the desired object in a multi-object scene. Our proposed strategy of shared cognitive control allows subjects with neuro-motor disabilities to grasp objects in their surroundings accurately and reliably with minimal cognitive effort.
- We propose a CNN based novel robotic grasp detection network named Overlapping Object Grasp Net(OOGNet), which is capable of grasping the desired object even if the object is partially overlapped by other objects. The proposed grasping model outperforms the baseline algorithms by a large margin.
- Here, the subject is relieved from planning a complex trajectory for the robot link to align it with the desired object as the entire reaching and grasping phase of the robot is made autonomous in the present paper. Hence, the proposed scheme requires very little subject training compared to existing state-of-the-art algorithms and reduces the overall workload of the subject.
- The proposed strategy significantly improves the success rate while minimizing the settling time and positional-steady error of the system. Autonomous navigation of the robot towards the desired object and the proposed OOGNet-based grasping strategy yielded superior performance.

In addition to this, the paper provides a comparative analysis of workload imposed on the performing subject while implementing different BCI schemes. Due to the limited cognitive processing ability of the human brain, workload analysis becomes a necessary method to evaluate the advantages of any BCI scheme over the others [46,47]. Such comparison also provides a tool of assessing the match between mental cost and system performance [48]. Here we adopt an NASA-TLX based workload analysis technique to compare our proposed BCI scheme with the two other state-of -the art BCI schemes that use manual trajectory planning [49]. NASA-TLX based workload assessment has greatly been adopted in the field of BCI since years and also proves to be an effective way of workload assessment [50,51].

2. System overview

Our setup consists of a 6-DOF robot arm (Model: ABB IRB 120), with a payload capability of 3 kg and a maximum reach of 3 ft, mounted beside a human subject. The links of this manipulator are connected by rotary joints allowing only rotational movements. A Microsoft kinect sensor(RGB camera with depth sensing device) is placed on the 5th axis of the robot arm, in such a way that one can change the camera's pan and tilt by rotating the 1st and 5th joints of the manipulator respectively. The kinect provides live RGB feedback of the surrounding environment to the subjects via an LCD monitor placed next to them. Multiple objects of different classes are arranged in a variety of layouts, in the vicinity of the robot arm with some objects overlapping with others. We aim to solve the task of locating, identifying and grasping the desired object (within the reach of the robot arm) with the minimal human intervention. A complete overview of the system is presented in Fig. 1. For the sake of simplification, we have taken the following liberties in our set-up:

- Although some objects may overlap with each other, each object is clearly visible from the initial position by changing the camera's Field of View (FOV).
- Different objects belonging to the same class are indistinguishable in nature.
- Grasping an object does not require re-arrangement of other overlying objects.

We have divided the problem into several sub-tasks.

1. ***Locating the object:*** Although the user can see the objects physically in the environment, all of them are not visible on the monitor as the Field-of-view(FOV) of the kinect is limited. So there may arise a need to move the kinect to bring the desired object into its FOV. We have thus developed an algorithm which allows the subject to use MI brain signals to change the kinect's pan and tilt based on the position of the desired object. The right and left arm motor imaginations are mapped to the clockwise and anti-clockwise movement respectively of the selected joint (1st/5th) , while feet imagery is used to toggle the selection between 1st and 5th joint. Hence a subject first uses feet imagery to select the desired joint followed by the left and right arm imagery to rotate the joint in desired direction. Present selection of the joint is displayed in the screen for convenience of control. Motor imageries are decoded by detecting each unique ERD/ERS MI pattern [19,20] using a CNN classifier.
   Once the object fully enters the kinect's FOV, the user stops the Motor imagination. If no MI pattern is detected in two consecutive time windows, the algorithm terminates, indicating to the system that the object has been successfully located. This stipulation reduces system sensitivity to both classification and user errors. The algorithm of the process is given in Algorithm 1.
2. ***Choosing the object:*** Depending on the layout, there may be multiple objects visible on the screen together with the desired one. In order to tell the system which object to grasp, we need to identify all the objects and select the desired one out of them. Our algorithm

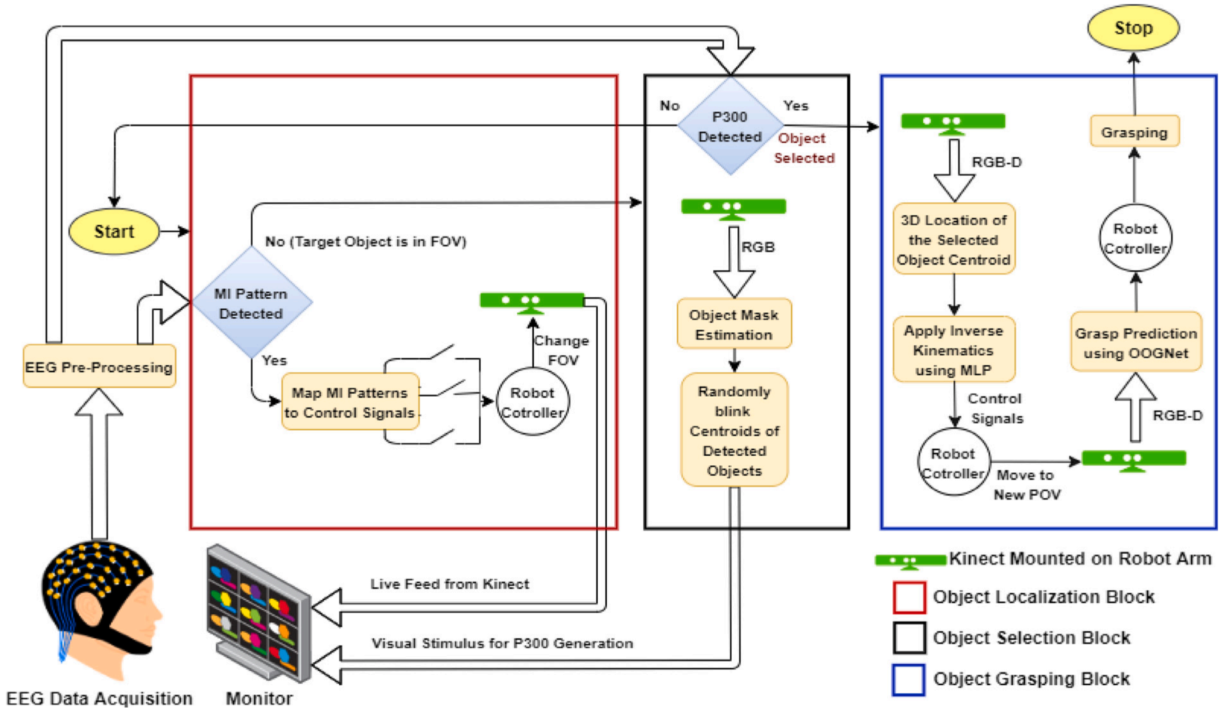


Fig. 1. Complete overview of the proposed scheme.

uses a state-of-the-art network Mask-RCNN [52] to detect objects in the image and calculates the object-centroids from their segmentation masks. The centroids are then flashed in a random sequence following the oddball paradigm [23]. When the desired centroid flashes, the subject gazing at that object elicits a P300 response which is detected by a CNN classifier to identify the target object for the system. The target object's class as predicted by Mask-RCNN is recorded for automatic identification in a later step. If no P300 signal is detected, the system returns to the starting state. Algorithm 2 explains the above procedure.

Algorithm 2: Algorithm for Object Selection

```
Perform Object Detection and Centroid Calculation;
Generate Random Sequence $S$ of Detected Objects;
for Objects in $S$ do
    Blink Centroids and Read EEG Data;
    if P300 detected then
        Choose Current Object;
        Exit Loop;
```

Algorithm 1: Algorithm for Object Localization

```
COUNT ← 1;
Default Joint ← 1st joint (Pan);
while $COUNT \leq 2$ do
    Provide Cue to Subject to Start MI;
    while Time within TIME WINDOW do
        Read EEG data;
    Provide Cue to Subject to Stop MI;
    Classify EEG for MI Tasks;
    if MI not detected then
        COUNT ← COUNT+1;
    else
        COUNT← 1;
        switch MI pattern do
            Feet: Toggle between Pan and Tilt;
            Left hand: Tilt Up/Pan Left;
            Right hand: Tilt Down/Pan Right;
```

3. ***Grasping the object:*** The system can now identify the desired object. But the end effector/gripper is still far away from its target to grasp it. We solve the automatic grasping problem in two steps:

   (a) ***Positioning:*** For any position (within the reach of the arm) of the desired object, the gripper first automatically moves closer to the object. Once the desired object is selected, the system calculates its real world 3D position from the 2D co-ordinates of the centroid and the depth map produced by the kinect. We apply inverse kinematics to move the gripper to a new position located a small fixed distance above the desired object, with the kinect tilted downwards to provide a new point of view (POV) from the top. This allows the next step to be independent of object position relative to the initial position of the gripper.

   (b) ***Estimating Gripper Configuration:*** The positioning step allows us to formulate the final task of grasping as the problem of calculating the gripper configuration from the object image. Jiang et al. [53] and Lenz et al. [54] have shown that a seven dimensional grasp orientation for a parallel plate gripper can be represented by a grasp rectangle parameterized by its position, width, height and orientation. So, the problem is reformulated as the task of predicting a grasp rectangle from the object's RGB-D image. In our case the top view may show multiple objects on the screen along with the desired one with possible overlapping layouts. We need to re-select the target object from the new POV as the previously calculated 3D position is not reliable in terms of identifying the object across different points of view of the camera, especially in case of close and overlapping object arrangements. Further, due to the proximity between objects, the problem cannot be reduced to a single object or a simple multi-object non-overlapping grasp detection

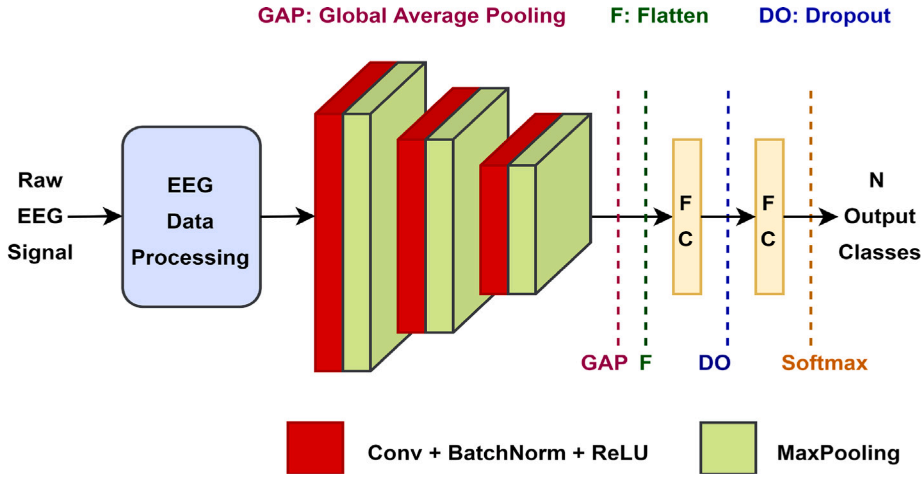


Fig. 2. The common EEG classification network architecture. Two separate instances of this network are used to classify MI and P300 signals with different network dimensions.

task like [53–60], as there arises a need for affiliation between object and grasp prediction.

Hence,our proposed network OOGNet generates a grasp rectangle, a bounding box and a class prediction for each object in the multi-object scene, thus ensuring a reliably high accuracy even in overlapping scenarios. Since objects of the same class are indistinguishable in our setup, the target object is re-identified automatically, by matching the object-classes predicted by OOGNet with the previously recorded class of the object chosen in selection stage. Once the target object is identified, the corresponding grasp rectangle is converted to a gripper configuration to grasp the desired object.

3. Method

3.1. EEG data analysis

The raw EEG data is usually noisy and contains a lot of irrelevant information. Hence, appropriate task-specific filtering techniques are applied to the EEG signal before classification.

3.1.1. Motor imagery

The EEG signal is band-pass filtered at 8–24 Hz to isolate the ERD/ERS phenomenon associated with MI brain patterns. Because of the effectiveness of Common Spatial Patterns (CSP) [61] in discriminating Motor Imagery tasks, we use CSP filters in a multiclass one vs rest scheme to project the EEG data along directions that maximize the differences between MI classes. The processed EEG signal (Y) is expressed as,

$$Y = WX \tag{1}$$

where, X is a Channels (C) × Time (T) matrix of the band-pass filtered EEG data and W is the $L \times T$ CSP projection matrix, with L spatial filters.

3.1.2. P300 ERP

P300 brain signal is characterized by a positive going peak around 300 ms after the onset of the target stimulus. A Chebyshev type I bandpass filter is used to filter the raw EEG signal between 1–10 Hz, to reduce the background noise. Next, the information relevant to the P300 ERP is isolated from the EEG data using Principal Component Analysis (PCA) [62]. PCA maps the signal to a lower dimensional space by extracting the K Eigenvectors from the EEG data that contain the most information for P300 responses. The lower dimensional signal S is expressed as,

$$S = PX \tag{2}$$

where, X is a Channels (C) × Time (T) matrix of the band-pass filtered EEG data and P contains the K eigenvectors as rows.

3.1.3. Feature extraction and classification

Convolutional Neural Networks (CNNs) have become extremely popular for EEG classification tasks due to their much higher accuracy compared to traditional linear classifiers. The processed 2D EEG data matrix is fed into 3 Convolutional and maxpooling layers to learn high level inferences from the data and a 1D feature vector is generated from the feature map by Global Average pooling and Flatten operations. The resulting vector is classified by a series of fully connected (FC) and dropout layers. The pooling and dropout operations prevent overfitting. Two different instances of CNN having same architectures and layers (as stated above) is used to classify Motor Imagery and P300 brain patterns respectively. The common CNN architecture used for the classification purpose is shown in Fig. 2. Minor adjustment is done where the final FC layer contains 3 neurons for MI classification and 2 neurons for P300 detection. For Motor Imagery, no MI pattern is detected if the probability of a particular EEG input does not exceed 0.5 for any of the classes.

3.2. Object detection and centroid calculation

In order to select our desired object, we first need to identify all the objects present in an image. Such object identification is carried out by a state-of-the-art object detection network, called Mask R-CNN [52] that can detect objects in a variety of closely positioned and overlapping layouts. For each input RGB image, the network predicts the class, bounding box and segmentation mask for every object visible in the image. We calculate the centroid (X_c, Y_c) of each object as,

$$X_c = \sum_i (X_i/n) \quad Y_c = \sum_i (Y_i/n) \tag{3}$$

where (X_i, Y_i) is the position of the ith pixel in its segmentation mask which contains n pixels in total.

3.3. Gripper alignment

The automatic positioning step allows us to reduce the cognitive load on the subject by making the system perform the precise position control task of reaching and grasping the desired object autonomously. The Kinect placed on the fifth axis of the robot arm, is equipped

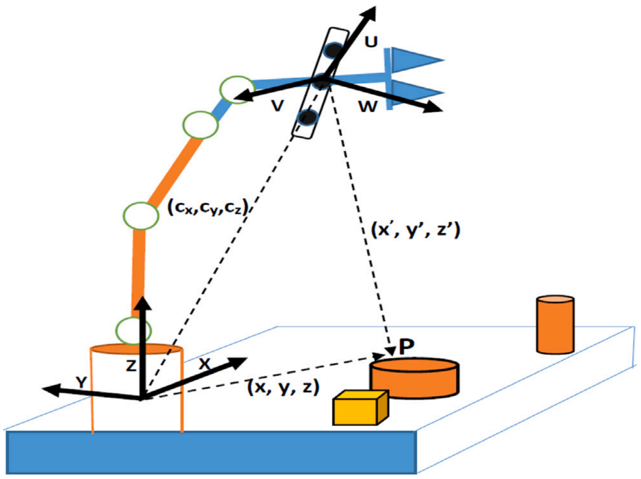


Fig. 3. Relative position of the object from kinect frame and robot base frame.

with an RGB camera and a depth sensor, which together provide the X and Y co-ordinates of the target object's centroid (obtained in the object detection and centroid calculation step) and its distance from the sensor. The centroid co-ordinates (x', y', z') measured from the Kinect frame are then transformed with reference to the base of the robot arm as shown in Fig. 3. The new co-ordinates of the centroid (x, y, z) with respect to the base frame can be obtained from Eq. (4).

$$\begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = \begin{pmatrix} R & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} I & c \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} \tag{4}$$

where, $\mathbf{c} = (c_x, c_y, c_z)$ is the position vector from origin of the base frame to the kinect frame and $R = [r_{ij}]$ is the rotation matrix, determined from the configuration (position and orientation) of the fifth link with reference to the base frame. Once the position of the centroid is known in terms of the base frame, the robot arm reaches the object in two stages.

In the first stage, the arm approaches the destination location $(x, y, z - \delta_z)$, where δ_z is a fixed offset distance above the centroid position along Z-axis. The robot arm employs an inverse kinematic model, which utilizes D-H parameters of the robot arm to find the required joint movements for reaching the destination. Since axes of the last three joints intersect at a point in the IRB 120 robot, only the first three joints contribute towards determining the position. For the destination position and the initial position of the end effector, $(x_d, y_d, z_d)^T$ and $(x_0, y_0, z_0)^T$ respectively, the required joint movements of the first three joints can be obtained from the following expression.

$$\begin{pmatrix} x_d \\ y_d \\ z_d \end{pmatrix} = {}^0Q_i \begin{pmatrix} x_0 \\ y_0 \\ z_0 \end{pmatrix} \tag{5}$$

where ${}^0Q_i = {}^0T_1.{}^1T_2...{}^{i-1}T_i$ and $i = 1, 2, 3$. Here ${}^{i-1}T_i$ is the transformation matrix for link i. Thus for a given destination co-ordinate, required joint movement is obtained by knowing the ${}^{i-1}T_i$ from below expression;

$$\begin{pmatrix} x_d \\ y_d \\ z_d \end{pmatrix} = \left({}^0T_1.{}^1T_2...{}^{i-2}T_{i-1}\right).{}^{i-1}T_i. \begin{pmatrix} x_0 \\ y_0 \\ z_0 \end{pmatrix} \tag{6}$$

Once the destination is reached, the 5th link is tilted downwards to orient the kinect in a fixed angular offset with the vertical, to obtain a top view of the desired object to calculate the gripper configuration using our grasp detection network.

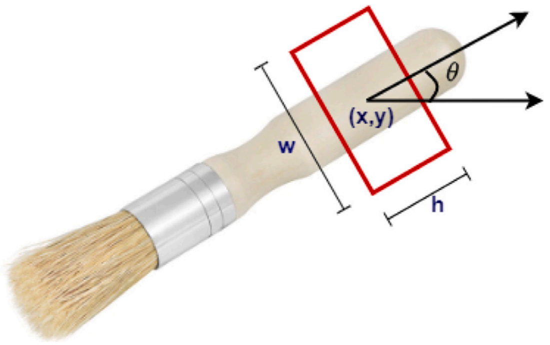


Fig. 4. 5-D rectangular representation of a gripper configuration. Here, (x,y) denotes the center of the grasp rectangle, w and h represent the width of the gripper opening and height of the gripper plates respectively and θ is the orientation of the grasp rectangle with respect to the horizontal direction.

After the grasp rectangle is predicted and the desired object is re-identified by its class, the second stage of approach is initiated. The co-ordinates (x_c, y_c, z_c) of the center of the grasp rectangle, calculated with respect to the base frame, is chosen as the grasping point. The gripper opening is fixed according to the rectangle width. Now, the orientation of the grasp rectangle determines the rotation of the gripper about the normal to the image plane (kinect axis). An illustrative picture of grasp rectangle is shown in Fig. 4. Now, the robot uses its inverse kinematics to achieve the particular orientation. Last three joint variables θ_4, θ_5, θ_6 is obtained by solving the rotation matrix ${}^3R_6$ in a similar manner described in Eq. (6), where ${}^3R_6 ={}^3 R_4{}^4R_5{}^5R_6$. With this configuration, the gripper approaches the object along the normal and grasps the object as described in [53,54].

3.4. Grasp detection

In our use-case, we need to predict grasps for multi-object overlapping scenes. This is significantly more difficult than non-overlapping or single object cases, due to partial occlusion by overlapping objects and the need for affiliation between object and predicted grasp. The grasping success rates of previous works [63–65] in this domain are too low for use in reliable human assistant systems. The absence of a repository for the cited implementations together with the availability of depth information at our disposal, motivated us to design a novel grasp prediction network, that would ensure a high physical grasping accuracy for our use-case. While the previous approaches use only RGB information, we use the RGB-D images from the kinect and a deeper feature extractor to improve accuracy. In order to maintain a fast enough execution speed for user-convenience, we predict only one grasp rectangle for each region of interest (ROI) instead of multiple rectangles unlike the previous methods. Our proposed Overlapping Object Grasping Network (OOGNet) generates a grasp rectangle, bounding box, and object class for each object in the image thus associating each predicted grasp with its object. Architecture of the OOGNet is shown in Fig. 5. Similar to [54], we represent a gripper configuration by a grasp rectangle (G) with 5 parameters as,

$$G = \{x, y, w, h, \theta\} \tag{7}$$

where (x, y) denotes the center of grasp rectangle, h denotes the height of parallel plates, w denotes the maximum distance between parallel plates and θ denotes the orientation of grasp rectangle with respect to the horizontal axis of the image.

4. Grasp detection network

Our proposed network takes an RGB-D image as input and generates multiple ROI proposals for objects present in the image. Each ROI is

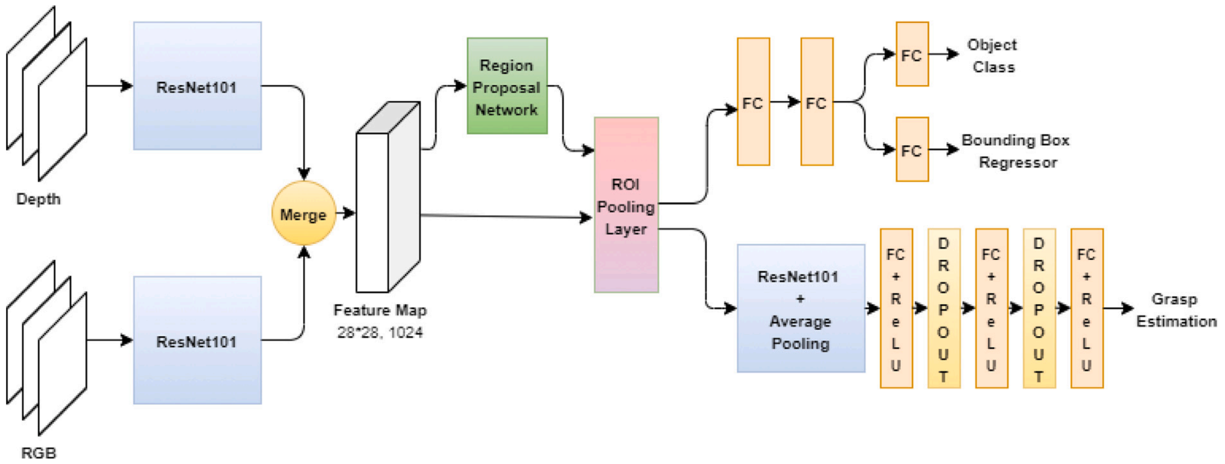


Fig. 5. Architecture of our proposed Overlapping Object Grasping Network (OOGNet). The network takes an RGB-D image of multiple overlapping objects as input and predicts the class, bounding box and a 5-D grasp rectangle for each object in the image.

then fed into three parallel branches that perform object classification, bounding box regression and grasp prediction. The object classifier and bounding box regressor branches are similar to that of Fast R-CNN [66] in structure. The grasp predictor branch regresses to the 5 parameters of a grasp rectangle for each object class. The following sections detail the architecture of our network in two stages that describe the generation of object ROIs from input image and the prediction of grasp rectangles from each ROI.

4.1. Object proposals

The first stage of the network generates object ROI proposals from the input RGB-D image. The single channel input depth image is converted to a 3-channel image by the grayscale to RGB conversion method. The 3-channel depth map and the RGB image are each fed into identical and parallel feature extractors. We use ResNet-101 [67] as the backbone of our feature extractor network. The skip connections in the Residual block allow us to use deeper networks that learn high level features without degradation of accuracy. The feature extractor in this stage contains the first 23 layers of ResNet-101. Feature maps of size 28 × 28 × 512 extracted from the depth and RGB inputs, are concatenated to form a merged feature map of size 28 × 28 × 1024. A Region Proposal Network (RPN) similar to that used by Faster R-CNN [68] is used to generate 9 (3 scales and 3 aspect ratios) Object ROI proposals for each location in the combined feature map. Each ROI is characterized by an objectness score (2 probabilities) and 4 parameters (x', y', w', h') denoting the bounding box location, where (x', y') specifies the top-left corner of the box and w' and h' denote width and height respectively. The RPN is trained in a similar fashion to [68] with the same loss function.

4.2. Grasp prediction branch

Each variable sized ROI generated from the RPN is fed into an ROI pooling layer together with the merged features to produce a smaller feature map of fixed spatial size (14 × 14). Three parallel branches share the pooled ROI feature map as input. The grasp branch contains a ResNet feature extractor that learns grasp specific inferences from the object ROIs. Here the feature extractor contains the last 51 layers of ResNet-101. The grasp feature maps of size $(7 \times 7 \times 2048)$ from the last convolutional layer of ResNet-101 are pooled by an average pooling layer and fed into three fully connected layers with ReLU activation. Each fully connected (FC) layer except the final one is followed by a dropout layer to reduce overfitting. The final FC layer outputs 5 × k grasp parameters, for the k object classes. Thus the grasp branch predicts a grasp rectangle for each class of object from the input object ROI.

4.3. Loss function

OOGnet generates three outputs, one from each branch. For each ROI, the classification branch predicts the softmax probabilities $p = (p_i | \forall i \in [0, k])$ of the object belonging to the k+1 classes; k object types and one background class denoting no object is present in the ROI. Here, p_i denote the softmax probability of the object belonging to class i. The bounding box and grasp branches regress to the bounding box parameters $t_i = (x'_i, y'_i, w'_i, h'_i)$ and the grasp parameters $G_i = (x_i, y_i, w_i, h_i, \theta_i)$ respectively for each of the k object classes, where i indexes the ith class.

The labels for each ROI include a ground truth class u, a ground truth bounding box regression target v and a ground truth grasp rectangle regression target g. Extending the loss in [66] we define a multitask loss L_{total} on each ROI to jointly train for classification, bounding box regression and grasp prediction.

$$\begin{aligned} L_{total}(p, u, v, t_i, g, G_i) = {} & L_{cls}(p, u) + \lambda[u \geq 1]L_{box}(t_u, v) \\ & + \lambda'[u \geq 1]L_{grasp}(G_u, g) \end{aligned} \tag{8}$$

Here, $L_{cls}(p, u) = -\log p_u$ is the classification loss. L_{box} is the bounding box loss defined over the predicted box parameters $t_u = (x'_u, y'_u, h'_u, w'_u)$ for the ground truth class u and the ground truth box parameter tuple v. The grasp loss L_{grasp} is added to the L_{cls} and L_{box} losses defined in Fast R-CNN [66] to simultaneously train for grasp predictions. L_{grasp} is defined over the ground truth grasp rectangle tuple $g = (g_x, g_u, g_w, g_h, g_\theta)$ and the predicted grasp rectangle $G_u = (x_u, y_u, w_u, h_u, \theta_u)$ for the groundtruth class u. Both L_{box} and L_{grasp} are smooth L_1 losses. For L_{grasp} the Smooth L_1 loss is expressed as,

$$L_{grasp}(G_u, g) = \sum_{j \in x,y,w,h,\theta} (smooth_{L_1}(G_u^j - g^j)) \tag{9}$$

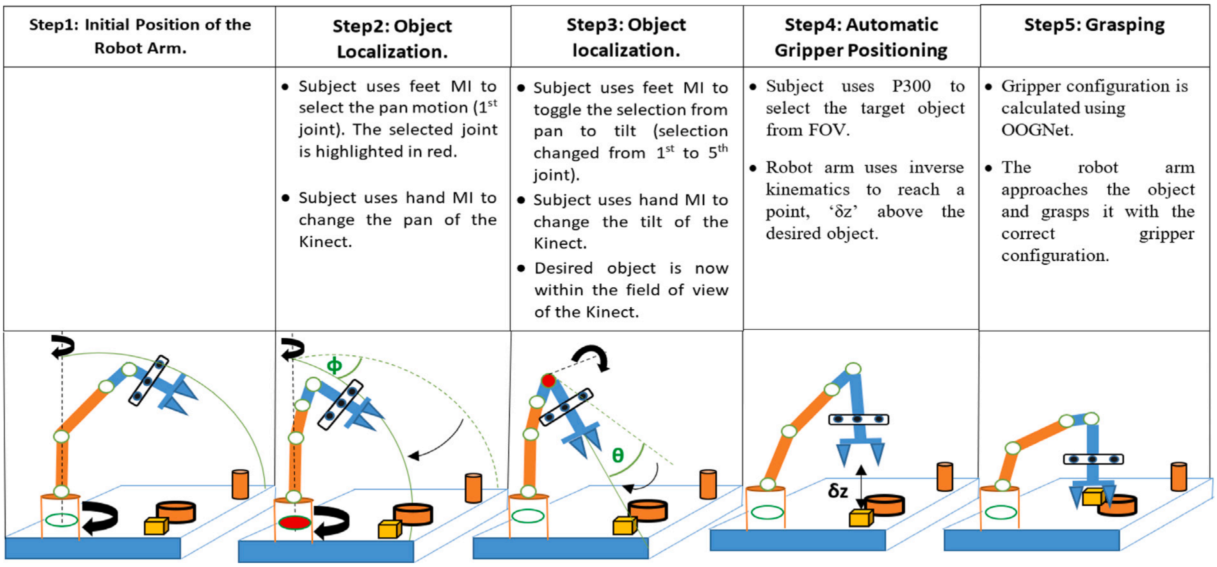
where

$$smooth_{L_1}(x) = \begin{cases} 0.5x^2, & \text{if } |x| \leq 1 \\ |x| - 0.5, & \text{otherwise,} \end{cases} \tag{10}$$

Smooth L_1 loss is used because of its robustness to outliers as pointed out by [66]. The L_{box} loss is defined similarly to L_{grasp}. The Iverson bracket indicator function $[u \geq 1]$ is defined as,

$$[u \geq 1] = \begin{cases} 1, & \text{if } u \geq 1 \\ 0, & \text{otherwise} \end{cases} \tag{11}$$

Labeling the background class as 0 together with the Iverson function allows the network to ignore the bounding box and grasp losses when the ROI is predicted to be background. This is essential as there is no object and hence no grasp able region in the image background. The



Step1: Initial Position of the Robot Arm.	Step2: Object Localization.	Step3: Object localization.	Step4: Automatic Gripper Positioning	Step5: Grasping
	• Subject uses feet MI to select the pan motion (1st joint). The selected joint is highlighted in red. • Subject uses hand MI to change the pan of the Kinect.	• Subject uses feet MI to toggle the selection from pan to tilt (selection changed from 1st to 5th joint). • Subject uses hand MI to change the tilt of the Kinect. • Desired object is now within the field of view of the Kinect.	• Subject uses P300 to select the target object from FOV. • Robot arm uses inverse kinematics to reach a point, 'δz' above the desired object.	• Gripper configuration is calculated using OOGNet. • The robot arm approaches the object and grasps it with the correct gripper configuration.

Fig. 6. Different stages of the robot manipulation in the testing session. Robot joints are marked with green circles while the currently selected joint is highlighted in red along with their axis of rotation. The dotted lines represent the axis of rotation.

hyper-parameters λ and λ' are the loss weights. They tune the relative weightages of the different loss terms. We use $\lambda = \lambda' = 1$ for our experiments. The ground-truth regression targets for both bounding box and grasp rectangle are also normalized to have zero mean and unit variance.

5. Experiments

5.1. Experimental protocol

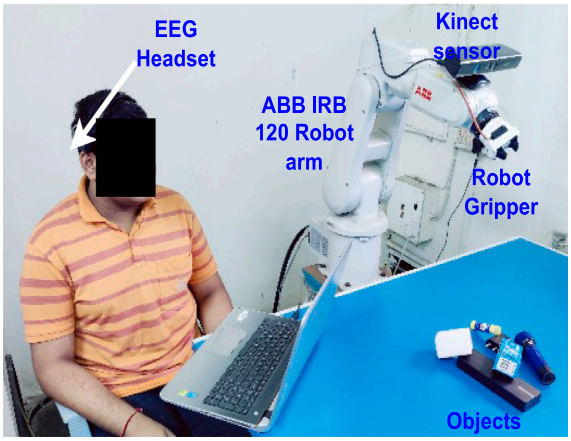
This section describes the experimental protocol employed in this study and highlights the key steps of conducting the experiment.

Subjects: The present study employs ten volunteers showing no major illness in their recent medical history. Out of ten volunteers six were male and four were female. All the volunteers belong to the age group of 18–35 with mean age of 30 years. The details of experiment and its objective were made clear to the all volunteers and a consent form, stating their interest to participate in the study, was duly signed by them. All the ethical and safety issues for employing human subject in the experiment is maintained according to Helsinki Declaration 1970 later revised in 2000 [69].

EEG System: EEG signal is acquired from the subjects using a 21 channel mobile EEG amplifier system. The amplifier has sampling rate of 200 Hz with built in notch filter at 50 Hz frequency. The present experiment follows the international 10–20 electrode positioning system to place the EEG electrodes in subjects scalp. Electrode position $C3, C4, Cz$ placed over the motor cortex region and $P3, P4, Pz$ placed over the parietal region are used to capture the motor imagery. The electrode position M1-M2(mastoid process) are used as contra-lateral referencing of all electrodes and Fpz is used as ground position. P300 brain pattern is captured from the electrode position Pz, Cz and Fz.

A figure depicting the different instruments used in the experiment is given in Fig. 7.

Communication Protocol: The EEG headset is wirelessly connected with a computer through Bluetooth protocol. The computer runs a python API to capture the EEG data and processes it in real-time, while an another computer(placed in front of the subject) runs a python script to capture and process the Kinect data in real-time. Both the computers are connected with a server computer using TCP/IP (creating TCP sockets in both server and client), where computers connected with EEG and Kinect act as the clients. The server computer generates the control commands for the robot based on the information provided by the computers connected with EEG and Kinect. The control algorithm in the server is executed in Robotstudio platform (by running a ABB Rapid language script) which again communicates with the physical robot controller(IRC 5 controller with Robotware version 6) using UDP (UDP socket). Once the generated control commands are sent to the robot controller, the robot joints are actuated and a specific task is executed.



[redacted]

Fig. 7. Depiction of experimental scenario and different instruments used in the experiment.

5.2. Training session

Training data for the classifier are obtained from ten subjects with the repetition of five sessions for each subject with inter-session gap of 30 min. Training session data are taken throughout the fifteen days. Each session contains thirty trials. Each of the trials contain visual cue of instructions to the subject. Timing diagram of the visual instruction is illustrated in Fig. 8. At the beginning of the trial a fixation cross appears in the visual cue for the 2s followed by a blank screen of 2s. Now a visual cue containing the instruction of motor imagery appears on the screen. Subject performs the motor imagery either to select the

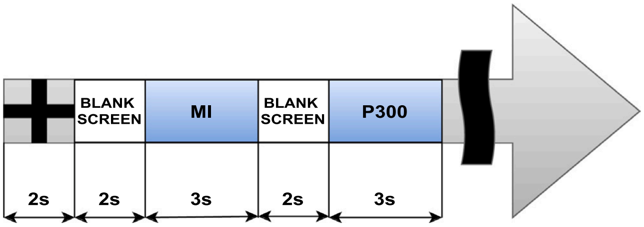


Fig. 8. Timing diagram of training session.

link (feet imagery) or to turn the virtual robot link in the given direction (left and right hand imagery). Hence, the camera mounted on the robot arm changes its FOV and now focuses on the objects on the table. Next the center points of the objects start blinking in random manner. A visual cue then appears on the screen to facilitate the P300 training data acquisition. The visual cue contains the instruction to the subjects to focus their gaze on a particular object. Subject develops the P300 brain response whenever the center point of that object is flashed.

5.3. Testing session

The testing session is more complex as no visual cue is shown and subject has to plan two steps of operation (MI generation and developing P300) without any assistance. During the testing session all the signal modalities are acquired with a moving window of 1s. An exception is followed in case of MI signal acquisition, where the signal is acquired for 1s but last 0.2 s of signal samples are considered for classification [70]. For better understanding the different stages of the operation is illustrated through sketches in Fig. 6 while the actual experimental scenario is depicted in Fig. 9.

6. Detailed experimental procedure of robotic grasp prediction

6.1. Datasets

There is no publicly available RGB-D dataset for robotic grasp detection in multi-object scenes. So, in order to train our model for overlapping multi-object scenes, we carefully collect a Multi-Object RGB-D dataset and annotate manually. For every object in a single image, we annotate several possible grasp rectangles which are a comprehensive subset of all possible good candidates. We take several images of the same set of objects with different orientation and pose. We also include the affiliation between each grasp with corresponding objects using the index of the object bounding boxes. Example images from multi-grasp dataset is shown in Fig. 10.

Our Multi-Object Grasping Dataset contains 784 images with 3–5 different objects in each image. The objects are arranged in several overlapping and non overlapping layouts. The dataset consists of both RGB and depth images. We use the same kinect as the depth sensor. There are in total of 17 classes and different instances of each class are indistinguishable in nature. The object bounding boxes and the grasp rectangles are manually annotated.

6.2. Pre-training and data pre-processing

Similar to [71], we reuse the pre-trained weights of ResNet-101 on ImageNet [72] dataset to avoid over-fitting. The new layer weights are randomly initialized with a zero-mean Gaussian distribution with standard deviation 0.02. The NaN values in the depth image are replaced with zeros. The depth image is converted to a 3-channel image using grayscale to RGB conversion method and is rescaled to the 0–255 range. As both datasets are small, we perform extensive data augmentations by randomly rotating, translating and changing the background color for regularization [73–75]. We also add noise, saturation, illumination and hue randomly, to make the system robust to real conditions.

6.3. Training

We train the entire network end-to-end using Pytorch framework on an NVIDIA GTX 1080 Ti GPU, with 16 GB dedicated memory, with CUDA-10 and cuDNN-7.5 installed. We randomly divide the Multi-Grasping dataset in 4:1 ratio for training and testing. There are 2016 object instances in training set and 758 object instances in the test set.

The training process is divided into two stages. First, we train the RPN using the input images and ground truth Object proposals as described in [68]. Next, the complete network is trained end-to-end. The pre-trained ResNet is fine tuned using stochastic gradient descent (SGD) optimizer with the hyper parameters set as: initial learning rate = 0.0001, mini batch size = 16, momentum = 0.9 and maximum number of epochs = 30. We divide the learning rate by 10 every 10000 iterations.

7. Results

7.1. Performance of EEG classifier

The performance of the proposed EEG classifier networks is evaluated on the basis of four metrics — Classification Accuracy (CA), True Positive Rate (TPR), False Positive Rate (FPR) and Cohen's kappa index (κ) which are defined as -

$$CA = \frac{TP + TN}{TP + TN + FP + FN} = p_a \tag{12}$$

$$TPR = \frac{TP}{TP + FN} = Sensitivity \tag{13}$$

$$FPR = \frac{FP}{FP + TN} = 1 - Specificity \tag{14}$$

$$\kappa = \frac{p_a - p_e}{1 - p_e} \tag{15}$$

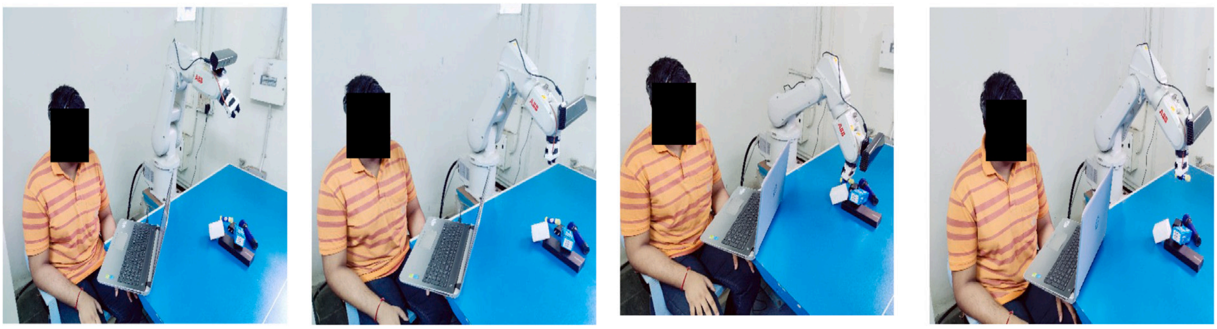
where, TP is the true positives, TN is the true negatives, FP is the false positives, FN is the false negatives, p_e is the chance of agreement that is expected and p_a is actual percentage of agreement. The random accuracy, p_e, is calculated as

$$p_e = \frac{(TN + FP)(TN + FN) + (FN + TP)(FP + TP)}{(TP + TN + FP + FN)^2} \tag{16}$$

The Classification Accuracy shows the percentage of trials in the test data that have been correctly classified. TPR and FPR shows the ability of the classifier to correctly detect the true positive and true negative instances out of total positive and negative instances respectively. Cohen's kappa index is a inter rater reliability measure of categorical items and it is used to assess the reliability of the classifier.

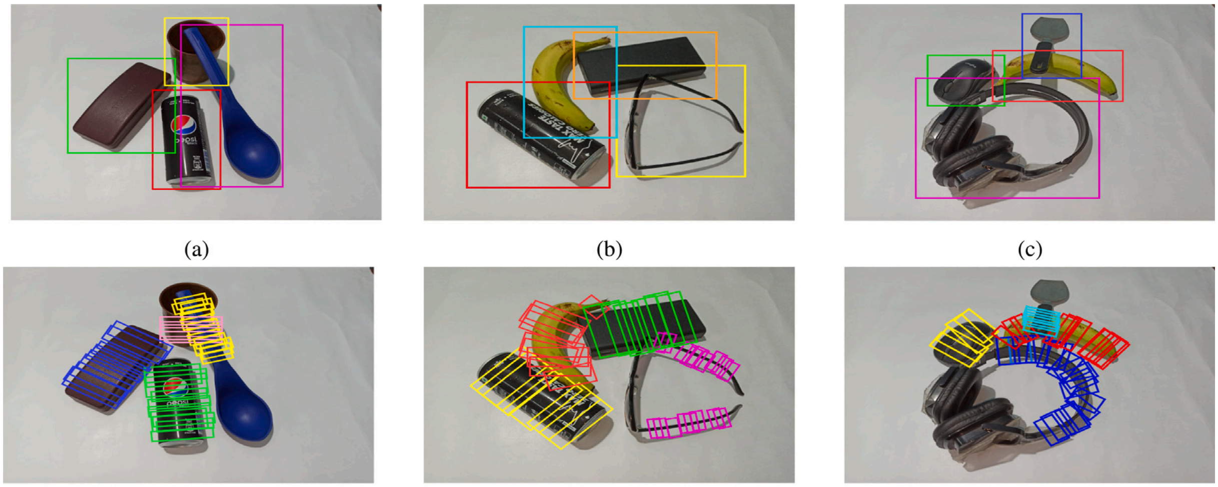
It can be seen from Table 1 that the proposed classifier outperforms both linear and non-linear classification methods for both MI and P300 classification. While a standard CNN works better than linear classifiers, pre-selection of features followed by a CNN achieves the best results with CA, TPR, FPR and κ values of 95.58%, 0.96, 0.05, 0.91 for MI and 96.30%, 0.91, 0.03 and 0.90 for P300. The better results for CSP and PCA compared to other techniques are expected as per our literature-supported intuitions about MI and P300 EEG respectively.

The metric values for 10 subjects have been reported in Table 2, for both MI and P300 classification, to highlight the inter-personal variance in the performance of our classifier. As we can see, the average values of CA, TPR, FPR and κ are 95.58%, 0.96, 0.05 and 0.91 respectively for MI classification, while for P300 detection, they are 96.3%, 0.91, 0.03 and 0.90 respectively. It is evident from the standard deviation values (written below the CA metric), that the inter-trial variation in the classifier performance is very small for both MI and P300 detection, indicating the high reliability and robustness of our proposed classifier, though P300 detection system shows more reliability than MI detection system.



(a) Step1 (b) Step2 (c) Step3 (d) Step4

Fig. 9. Different steps of experimental scenario where the participating subject is mentally guiding the robot to reach and grasp the desired object.
Step 1: The robot is at it is initial position and the subject uses his motor imagery and P300 to rotate the robot arm and select the desired object respectively. Step 2: The robot is just above the desired object and determines the gripper configuration. Step 3: The robot has successfully grasped the desired object. Step 4: Grasped object is picked up by the robot to place it in other place.



(a) (b) (c)

(d) (e) (f)

Fig. 10. Example images from multi-grasp dataset.

Table 1
Comparative study of different EEG classifiers.

	Classifiers with Optional	Performance metrics			
	EEG Pre-processing	CA (%)	TPR	FPR	κ
MI classifier	LSVM [76]	85.80	0.85	0.07	0.83
	KSVM-RBF Kernel [77]	87.55	0.86	0.06	0.86
	IT2FS [78]	90.54	0.88	0.06	0.88
	GT2FS [79]	90.65	0.89	0.04	0.87
	BPNN [80]	89.82	0.86	0.08	0.82
	CNN [81]	92.26	0.92	0.04	0.89
	STFT [82] + CNN	94.32	0.94	0.05	0.90
	DWT [83] + CNN	94.75	0.95	0.04	0.90
	CSP [61] + CNN	**95.58**	**0.96**	**0.05**	**0.91**
P300 classifier	SWLDA [84]	90.23	0.84	0.07	0.83
	LSVM [76]	90.81	0.86	0.05	0.86
	KSVM-RBF Kernel [77]	92.56	0.90	0.05	0.88
	BPNN [80]	89.80	0.84	0.04	0.86
	CNN [81]	93.95	0.90	0.04	0.90
	ICA [85] + CNN	95.12	0.91	0.04	0.90
	MRMR [86] + CNN	94.05	0.90	0.03	0.89
	PCA [62] + CNN	**96.30**	**0.91**	**0.03**	**0.90**

7.2. Statistical validation of the classifiers

Classifiers are statistically validated using Friedman statistical test. The Friedman test is a non-parametric test (does not hold the assumption that the data come from a normal distribution) that determines if there exists any significant difference between the classifier performance based on any selected parameter and ranks them according to it. Here, we have considered two different parameters, Accuracy and Reliability(kappa score) and performed the Friedman test separately for each of these parameters. The test considers a null-hypothesis that assumes the performance of the classifiers under testing is equal based on the selected parameter, hence the sum of their ranks, which are assigned based on their performance, are also equal. Under the null hypothesis Friedman statistic is distributed as χ with $n-1$ degrees of freedom, where n is the number of classifiers under testing. Mathematically the Friedman statistic is computed as below;

$$\chi_F^2 = \frac{12}{Ln(n+1)} \sum_{i=1}^{n} R_i^2 - 3L(n+1) \tag{17}$$

where L is the number of data-set (we considered data-set averaged over all the sessions for each of the participating subject, hence L=10), n is the number of classifiers under testing and R_i is the rank sum of the classifier which was determined by summing all the ranks it got from all the data-sets based on the performance on that data-set. The values of χ_F^2 is obtained separately for each category of signal (MI and P300) and

Table 2
Performance of the proposed EEG classifier algorithms in multiple runs for different subjects.

	Metric	Sub 1	Sub 2	Sub 3	Sub 4	Sub 5	Sub 6	Sub 7	Sub 8	Sub 9	Sub 10	Avg
MI Classifier CSP + CNN	CA(%)	96.75 (±1.31)	96.10 (±1.37)	94.05 (±2.21)	97.05 (±1.22)	92.85 (±0.85)	94.48 (±1.30)	95.90 (±2.28)	95.25 (±0.81)	97.59 (±0.32)	95.92 (±0.62)	95.58
	TPR	0.97	0.96	0.94	0.98	0.92	0.95	0.97	0.98	0.98	0.94	0.96
	FPR	0.03	0.04	0.06	0.04	0.06	0.06	0.05	0.07	0.03	0.03	0.05
	κ	0.93	0.92	0.88	0.94	0.86	0.89	0.92	0.90	0.95	0.92	0.91
	Time (s)	0.591	0.594	0.564	0.570	0.544	0.585	0.580	0.576	0.590	0.568	0.576
P300 Classifier PCA + CNN	CA(%)	95.0 (±0.21)	97.58 (±0.15)	95.8 (±0.44)	96.3 (±0.31)	98.1 (±0.25)	96.9 (±0.12)	94.7 (±0.25)	97.8 (±0.29)	93.9 (±0.40)	97.4 (±0.30)	96.3
	TPR	0.92	0.96	0.89	0.88	0.94	0.90	0.89	0.95	0.83	0.92	0.91
	FPR	0.04	0.02	0.03	0.02	0.01	0.03	0.04	0.02	0.05	0.01	0.03
	κ	0.82	0.94	0.89	0.94	0.98	0.90	0.83	0.96	0.80	0.96	0.90
	Time (s)	0.121	0.186	0.152	0.162	0.140	0.180	0.156	0.126	0.134	0.160	0.151

Table 3
Results of Friedman statistical test.

Category	Parameter	χ_F^2 value obtained from test	Critical χ_F^2 value	Null Hypothesis Accepted/Rejected
MI	Accuracy	71.52	16.91	Rejected
	Kappa	61.66		Rejected
P300	Accuracy	65.13	15.50	Rejected
	Kappa	33.03		Rejected

compared with the critical value of the χ_F^2 ($\alpha = 0.95$). If the obtained value crosses the critical value, we conclude that a significant difference exists between the performance of the classifiers and the classifiers can be ranked as per the cumulative rank sum. The classifier having the lowest cumulative sum is considered as the best performing classifier.

MI classifier validation: During the MI classification process, performance of the proposed classifier is compared with eight other classifiers, hence we consider n=9 and L=10 in this case. The statistical test is carried out in two phases, in the first phase we ranked the performance of the classifiers based on accuracy and in the second phase we ranked them based on kappa score. Cumulative sum of the ranks are obtained and put into (17) separately for two cases and in each case the obtained χ_F^2 value exceeds the critical value. Detailed results are given in Table 3.

P300 classifier validation: Performance of the proposed P300 classifier is evaluated over 10 data-sets (L=10) and compared with seven other classifiers (n=8). The statistical test is carried out in the same manner as described above. The result is given in Table 3. It is evident from the result that obtained χ_F^2 value exceeds the critical value in each cases.

As the obtained χ_F^2 value exceeds the corresponding critical value in every cases, we conclude that null hypothesis is rejected in each case. Hence, the performance of the classifiers can be evaluated by their cumulative ranks and the classifier with the lowest rank has the best performance.

In the MI classification process, our proposed classifier achieved lowest cumulative ranks of 13 for both the accuracy and kappa score. In case of P300 classification process, our proposed classifier got the lowest cumulative ranks of 13 and 12 for accuracy and kappa score respectively. Hence, in each case the proposed classifier performs best among others.

7.3. Performance of grasp prediction network

In order to evaluate the performance of our model on the Multi-Object Grasping Dataset for object overlapping scenes, we need to take into account both object detection and grasp rectangle regression performances, since object-grasp affiliation requires accurate classification and localization of the objects in the image. For our task, we use an mAP based metric called mAPg defined in [63–65]. A detected object-grasp pair is labeled successful if:

Table 4
Evaluation on multi-grasp dataset.

Algorithms	mAPg (%)	Speed (fps)
Faster-RCNN [68] (RGB) + GR-ConvNet [58] (RGB-D)	72.1	**46.9**
Faster-RCNN [68] (RGB) + (ResNet-50) FCGN [59] (RGB)	64.5	11.9
Faster-RCNN [68] (RGB)+ (ResNet-50) FCGN [59] (RGD)	63.3	11.9
Faster-RCNN [68] (RGB) + (ResNet-101) FCGN [59] (RGB)	69.5	10.2
Faster-RCNN [68] (RGB) + (ResNet-101) FCGN [59] (RGD)	68.2	10.2
OOGNet (RGB-D)	**80.4**	11.1

1. the object is classified correctly and the predicted object bounding box has an IOU higher than 0.5 with the ground truth bounding box
2. the detected grasp is labeled as a good grasp according to the rectangular metric defined in [53], subject to the following criteria:
    - the difference between predicted grasp angle (orientation) and ground truth grasp angle is less than 30°
    - the Jaccard Index(J) between ground truth grasp rectangle (g) and predicted grasp rectangle (G), as defined below, is more than 0.25.

$$J(G,g) = \frac{(g \cap G)}{(g \cup G)} \tag{18}$$

1. **Performance on Multi grasp dataset:** The previous works [63–65] on simultaneous object detection and grasp prediction have not made their code or architectural details public, ruling out any possibility of re-creation. While they have reported mAP scores on the VMRD dataset defined in [87], the absence of depth data prevents us from evaluating our network on VMRD. So, in order to provide proper context to the performance of our network in overlapping object scenes, we select a combination of a state of the art object detector, Faster-RCNN [68] and two state of the art grasp detection models, GR-Convnet [58] and FCGN [59] including all the architecture variations of the latter. Since these combined networks have no implicit grasp-object affiliation, the grasp rectangle with confidence score higher than 0.25 and center closest to the object bounding box center is associated with each detected object. Since our dataset contains RGB-D images, we evaluate the FCGN model, equipped for 3

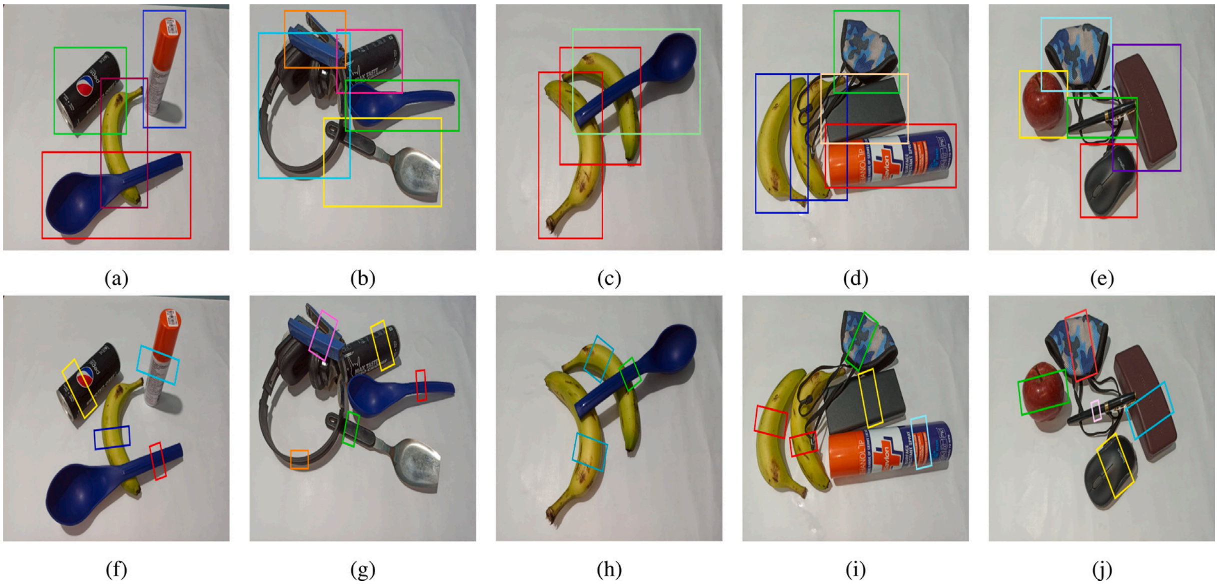


Fig. 11. Example of detection results on the multi-grasp dataset. The top row shows the predicted bounding boxes and the bottom row shows the predicted grasps on each corresponding test images.

Table 5
Results of Physical Grasping Experiments.

Objects	Grasp Prediction Success rate		Grasp Execution Success rate	
	Single	Multiple	Single	Multiple
Banana	10/10	10/10	10/10	10/10
Headphone	10/10	10/10	9/10	9/10
Can	10/10	10/10	10/10	10/10
Stapler	10/10	10/10	10/10	10/10
Spectacle	10/10	10/10	10/10	10/10
Spoon	10/10	9/10	10/10	9/10
Box	10/10	10/10	10/10	10/10
Mask	9/10	9/10	9/10	8/10
Apple	10/10	10/10	10/10	10/10
Mouse	10/10	10/10	10/10	10/10
Torch	9/10	9/10	9/10	10/10
Glue Stick	10/10	10/10	10/10	10/10
Mobile Charger	10/10	9/10	10/10	9/10
Battery Box	10/10	10/10	10/10	10/10
Total	**98.6%**	**97.1%**	**97.8%**	**96.4%**

Table 6
Performance comparison the proposed system with existing hybrid closed loop BCI schemes.

Performance metric	MI+ErrP [34]	SSVEP+MI+P300 [33]	Proposed Method
Success rate (%)	85.6	90.2	**93.4**
Steady state error (%)	2.1	0.2	**0.05**
Settling Time (s)	31	24	**15.92**
Peak Overshoot (%)	4.9	4.2	**0**

channel inputs on both RGB and RGD modalities while the GR-Convnet is evaluated on RGB-D as intended by its authors. Table 4 demonstrates that our model outperforms the other models by a large margin with an mAP score of 80.4%. Although the GR-ConvNet is shown to be quite effective at handling cluttered object scenes, our network still beats it in terms of performance. The improvement can be explained in part by the increased effectiveness of our model in dealing with the partial occlusion of objects in case of overlapping layouts, and in part by the simplistic object-grasp affiliation scheme used in the absence of an implicit association in case of the combination models. While FCGN has a higher mAP for RGB input compared to RGD, the mAP of the FCGN model (69.5%) has also increased from that reported by [63] on VMRD (54.5%). The reason for this increase may be attributed to the fact that the overlapping object layouts in our Multi grasp dataset are relatively less complex than that in VMRD due to our precondition for object visibility as stated in Section 2. An execution speed of 11.1 fps for our model is more than sufficient for user convenience. The improvement in speed over the FCGN model, in spite of having a deeper feature extractor, can be explained by our choice to generate only object proposals instead of predicting multiple object and grasp candidates. Performance of the grasp prediction of the OOGNet is represented with few example images in Fig. 11.

2. **Physical Evaluation:** In order to ascertain how well the grasp predictions translate to successful physical grasps in real-world, we perform extensive experiments. In experimental set-up, the robot arm is positioned a fixed distance above a plane surface containing either a single object or multiple objects in a variety of overlapping layouts, with the kinect mounted on its 5th axis and tilted downwards. Our network takes an RGB-D image of the top view of the objects as input from the kinect and generates a class prediction, bounding box and grasp rectangle for each object in the image. For multi-object layout, a particular object is selected to be grasped. The predicted grasp rectangle is converted to a gripper pose; the arm approaches and grasps the target object as described in [53,54]. The experiment is performed on 10 different objects, with 25 trials for grasping each object in both single and multi-object overlapping layouts.

Table 5 shows the success rates for both the grasp prediction and execution for each object in single and Multi-object scenes in a benchmark scale of 10 as used in [52][55]. For single object cases, our model reached a 98.6% prediction success rate and a 97.8% success rate for physical grasping over all objects. On the other hand for Multi-object scenes, our network achieved a staggering 96.4% success rate for physical grasping and a 97.1% success rate for prediction. The results are calculated over 10 trials chosen randomly from the original 25 trials, to remove any biases present in the manual arrangement or positioning of the objects in single and multi-object settings. This highlights the effectiveness and reliability of our proposed network in performing real-world grasping tasks for a variety of object arrangements.

Table 7
Online performance results.

Sub	Method 1 [34]		Method 2 [33]		Proposed Method	
	Acc_{BCI}	Acc_{Sys}	Acc_{BCI}	Acc_{Sys}	Acc_{BCI}	Acc_{Sys}
S1	79.16	68.75	89.58	83.33	85.41	85.41
S2	85.41	79.16	81.25	81.25	91.66	87.5
S3	83.33	72.91	85.41	83.33	87.50	85.41
S4	85.41	79.16	87.50	79.16	91.66	89.58
S5	85.41	72.91	83.33	81.25	93.75	91.66
S6	87.50	79.16	75.00	72.91	95.83	91.66
S7	79.16	68.75	62.50	58.33	85.41	83.33
S8	83.33	72.91	81.25	75.00	89.58	87.50
S9	75.00	66.66	72.91	70.83	83.33	79.16
S10	81.25	70.83	85.41	79.16	93.75	89.58
Avg	82.50	73.12	80.41	76.45	89.79	87.08

7.4. System performance

The overall position control performance of the BCI system is assessed here using few popular metrics arrived from control system literature. The metrics viz. success rate, settling time, peak overshoot and steady state error are considered to evaluate the system performance. Formal definition of the metrics are given below.

Success rate: It expresses the number of successful attempts out of the total attempts made by the robot to reach the desired object. An attempt is regarded as successful only when the robot is able to grasp the desired object properly.

Steady state error: It indicates the maximum positional deviation of the robot end effector from the desired position in the infinite time range.

Settling Time: Time taken by the system to reach and stay within 2% of steady state position.

Peak Overshoot: The maximum deviation of the response from its desired position. It is expressed as percentage change from its final response.

Performance of the overall system is given in Table 6 and the online performance is provided in Table 7. In both the cases, result is also compared with the performance of the two recent state of the art work [33,34] which fall under the category of hybrid closed loop BCI and employs manual trajectory planning. Table 6 reports the overall success rate of the system by counting the number of time the subject is able to reach the desired position, however the subjects are allowed to retake their decision if their intent is miss-classified at any stage. Table 6 focuses more on how the proposed control and planning method effects the overall system performance. On the other hand, Table 7 is obtained by following an online protocol which rejects the entire trial if miss-classification occurs at any stage of a trial and considers a trial to be successful only if all the stages of it are successful. Accuracy of the BCI and overall system are reported separately to provide insight to the readers how the BCI performance effects the overall system accuracy.

Time taken by the each module of the present work and their individual success rates are reported in Table 8. The last row denoting total system performance, provides the average performance of the entire system when all the modules work together, which is not equal to the numerical average of the each module. The above table also indicates the human involvement in each of the module. Hence the time taken by the first two modules is greatly effected by human behavior where as the time taken by the last two modules is affected by velocity of the robot arm and shape of the object selected by the subject. Subcomponents of system that govern the real time behavior of the system are described below along with the execution time. Object localization module includes detection of Motor signals that requires approximately 0.38 s including signal acquisition for 0.3s and classification time of 0.08 s. Robot actuation time is 0.01 s. Rest of the time is accounted for the control of the robot arm by the human subject to select the desired FOV. Next, the object selection step aims to choose the target object when multiple objects are present in the FOV. This includes object detection using Mask RCNN (0.1 s), Centroid calculation (< 0.01 s) and P300 detection (0.58 s= signal acquisition for 0.5s+classification time 0.08 s). Rest of the time is taken by the human subject to decide which object he/she wants to choose. The Positioning step aims to estimate the 3D coordinates of the target object using a 2-D spatial location and a 2-D depth map. Co-ordinate estimation takes an average of 0.05s. Reaching the estimated position by the robot arm depends on end effector velocity and distance between present and estimated position. Lastly, the final gripper configuration is estimated using our proposed OOGNet architecture. This module takes approximately 0.1 s (11 fps) and gripper actuation time is 0.01 s The total gripping time depends on the size and shape of the object selected by human.

It is apparent from the Table 6 that success rate of the proposed method is increased in significant margin of 3.2% and from the traditional BCI based success rate reported previously [33]. The overall success rate is found to be 93.4%. The steady state error is also drastically reduced to 0.05% along with the settling time which is further reduced to 15.92s. The proposed method shows no overshoot or undershoot in either of the experiment due to the over-damped response of the robot end effector. Absence of human involvement in the gripper positioning phase and invoking autonomous positioning module have eliminated the oscillation of robot end effector around the targeted object, which is otherwise reported in existing literature. A similar trend is seen in online protocol results reported in Table 7 where the best overall system accuracy and BCI accuracy are found to be 87.08% and 89.79% respectively for the proposed method. Such increase in BCI and overall accuracy may be attributed to the fact that the proposed method uses minimal human intervention hence minimizes the error that may arise from BCI decoding performance. The judicious selection of autonomous positioning and grasping strategy also contributed to make the system more robust and highly accurate compared to other methods reported in the table.

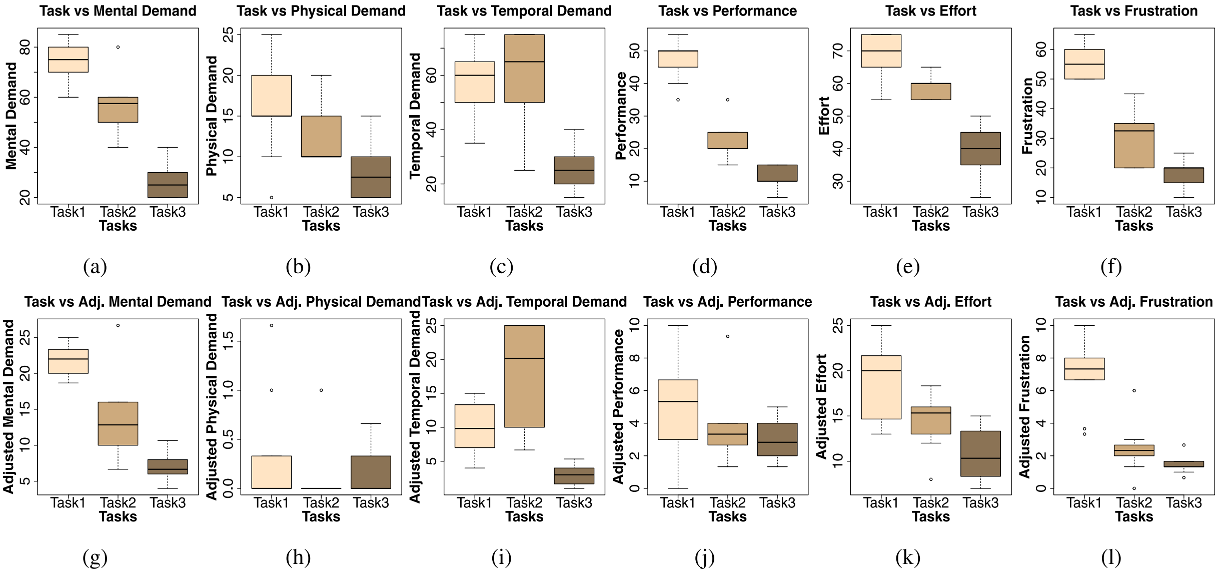
As per Table 8, average success rate of the individual modules are found to be 94.1%, 95.0%, 99.8% and 96.2% for Object localization, Object selection, Automatic positioning and Grasping phase respectively. It is also noticed that when subject performs each module consecutively in a single run, the overall success rate of the system slightly reduces to 93.4% which is less than the numerical average of individual success rates.

Removing human interaction from the end-effector positioning and grasping phase has a significant effect on reducing workload of the subject during real time operation. The complex planning procedure of aligning robot end effector with desired object imposes a heavy workload on the subject. It also requires a significant amount of subject training and most of the novice subjects are not able to do it with required accuracy. Experiments [33,34] involving such complex planning procedure are replicated in the laboratory environment and workload of the subject while operating under those schemes is compared with the present proposed method. The main motivation behind providing this comparison study is to indicate the quantitative difference in workload associated with pure cognitive control based state-of-the art BCI approaches(subject has to plan the entire robot trajectory for reaching and grasping) and our proposed scheme (shared control based approach where reaching and grasping phase were made autonomous,minimizing overall human intervention).

Overall Workload assessment: Workload of the participating subjects were analyzed using NASA-TLX questionnaire survey developed by NASA Ames Research Center that allows to asses the workload of the subjects operating various human–machine systems [49]. It assess the workload using a multidimensional rating system with six sub-scales: Mental Demands, Physical Demands,Temporal Demands, Performance, Effort, and Frustration [50,51]. Each sub-scale is divided into 20 equal

Table 8
Comparison between different modules of the proposed system.

Proposed module	Average execution time (s)	Average success rate (%)	Human involvement
Object Localization	7.18	94.1	Yes
Object Selection	2.85	95.0	Yes
Automatic Positioning	3.22	99.8	No
Grasping	2.64	96.2	No
Overall System Performance	15.92	93.4	–



(a) (b) (c) (d) (e) (f)

(g) (h) (i) (j) (k) (l)

Fig. 12. Box plot of sub-scales of the NASA-TLX study reported by ten participating subjects. The upper row represents the raw TLX scores whereas the lower row represents the adjusted TLX scores.

intervals which represents the score 0–100. Subjects provide rating over each sub-scale for the task they were assigned. Here three BCI systems were compared, hence the subject provided the rating for three tasks. Once the subject finishes the rating, 15 pairwise comparison between the sub-scales is presented to them, where subjects need to choose the sub-scale contributed most to their workload. Weight of a sub-scale is determined by number of time it is chosen by the subject during pair wise comparison task. The overall score of the test is found by computing the weighted average of the sub-scales with the weights determined above.

Here 10 participating subjects provided the rating for six sub-scales for each of three BCI tasks. Hence a total of 180 responses ($10 \times 6 \times 3$) were recorded. Average ratings of each BCI tasks for the six sub-scales are found by averaging the response over ten subjects. The result is shown in Fig. 12. Adjusted rating of each sub-scale is also reported in the above figure. RAW TLX scores reported in the first row of the figure reveals that Task 1 [34] imposed highest mental load and physical load on the subject, whereas Task 2 [33] imposes highest temporal load. Task 3 (proposed strategy) has been the lowest in all of the above categories and also demands least effort from the subject to operate it. Task 3 is also found to have the highest performance rating and lowest frustration rating. Adjusted TLX scores show similar pattern except for physical demand where it is found to be negligible in all three tasks. The overall adjusted TLX scores of Task1 and Task 2 are found to be $62.49 \pm 4.59(mean \pm std)$ and 51.43 ± 5.58 respectively whereas the overall score of the Task3 is found to be 23.99 ± 6.80 imposing least cognitive load on the participating subjects compared to the other manual trajectory planning based state-of-the-art BCI robot manipulation techniques.

8. Conclusion

Main motivation of the present work was to relieve the subjects from manual complex trajectory planning of the robot arm in a BCI based robot control scheme. The complex trajectory planning is mainly involved in object reaching and grasping task, which are made autonomous in the present scheme. The idea facilitates the precise grasping of any mentally selected object without any human intervention hence reducing the cognitive load of the subject drastically. The paper also proposed a CNN based novel robotic grasp detection network to predict the accurate grasp in real time. The proposed network is able to work on overlapping scenes and uses simultaneous object and grasp detection. The overall performance of the BCI system is greatly improved from the recent state-of the art where trajectory planning is entirely done by human subject. As an example steady state error of the system is reduced to 0.05% and settling time is reduced to 15.92 s. The results are substantiated by providing a comparison of cognitive load of participating subject for the proposed scheme and other recent BCI schemes. It was evident from the comparison that present scheme imposes least cognitive load on the subject and hence more suitable than the scheme involving manual trajectory planning. However there exists an ample scope to further reduce the cognitive load of the subject by suitably predicting the human behavior of selecting any object at any stage of operation, and assisting the human with autonomous navigational commands to reach the desired object. Such learning mechanism will reduce the need of multiple P300 generation that is used here for selecting the desired object. Such scheme uses fewer number of mental commands hence decreases the cognitive load and simultaneously increases the real-time accuracy of the system.

CRediT authorship contribution statement

Arnab Rakshit: Conceptualization, Formal analysis, Investigation, Methodology, Software, Writing – original draft. **Shraman Pramanick:** Formal analysis, Methodology, Software, Writing – original draft. **Anurag Bagchi:** Formal analysis, Methodology, Software, Writing – original draft. **Saugat Bhattacharyya:** Supervision.

Declaration of competing interest

No author associated with this paper has disclosed any potential or pertinent conflicts which may be perceived to have impending conflict with this work. For full disclosure statements refer to https://doi.org/10.1016/j.bspc.2023.104765.

Data availability

Data will be made available on request.

Acknowledgments

We thank Prof. Amit Konar, Head of the Artificial Intelligence lab, Jadavpur University, India, for providing us the necessary setup and facilities to conduct the experiment. Authors are also thankful to Laishram Rokesh Singh, (CAE Engineer, ARTPARK) for providing us with the robot gripper.

References

[1] Jonathan R. Wolpaw, Niels Birbaumer, Dennis J. McFarland, Gert Pfurtscheller, Theresa M. Vaughan, Brain–computer interfaces for communication and control, Clin. Neurophysiol. (2002).

[2] Feng Duan, Dongxue Lin, Wenyu Li, Zhao Zhang, Design of a multimodal EEG-based hybrid BCI system with visual servo module, IEEE Trans. Auton. Ment. Dev. (2015).

[3] Javier Andreu-Perez, Fan Cao, Hani Hagras, Guang-Zhong Yang, A self-adaptive online brain–machine interface of a humanoid robot through a general type-2 fuzzy inference system, IEEE Trans. Fuzzy Syst. (2016).

[4] Alexander J. Doud, John P. Lucas, Marc T. Pisansky, Bin He, Continuous three-dimensional control of a virtual helicopter using a motor imagery based brain-computer interface, PLoS One (2011).

[5] Tianwei Shi, Hong Wang, Chi Zhang, Brain Computer Interface system based on indoor semi-autonomous navigation and motor imagery for Unmanned Aerial Vehicle control, Expert Syst. Appl. (2015).

[6] Keun-Tae Kim, Heung-Il Suk, Seong-Whan Lee, Commanding a brain-controlled wheelchair using steady-state somatosensory evoked potentials, IEEE Trans. Neural Syst. Rehabil. Eng. (2016).

[7] Zhijun Li, Suna Zhao, Jiding Duan, Chun-Yi Su, Chenguang Yang, Xingang Zhao, Human cooperative wheelchair with brain–machine interaction based on shared control strategy, IEEE/ASME Trans. Mechatronics (2016).

[8] Carlos Escolano, Javier Mauricio Antelis, Javier Minguez, A telepresence mobile robot controlled with a noninvasive brain–computer interface, IEEE Trans. Syst. Man Cybern. B (2011).

[9] Suna Zhao, Zhijun Li, Rongxin Cui, Yu Kang, Fuchun Sun, Rong Song, Brain–machine interfacing-based teleoperation of multiple coordinated mobile robots, IEEE Trans. Ind. Electron. (2016).

[10] Ang Kai Keng, Guan Cuntai, Brain-computer interface in stroke rehabilitation, J. Comput. Sci. Eng. (2013).

[11] Nikhil Sharma, Valerie M. Pomeroy, Jean-Claude Baron, Motor imagery: a backdoor to the motor system after stroke? Stroke (2006).

[12] Nicholas Cheng, Kok Soon Phua, Hwa Sen Lai, Pui Kit Tam, Ka Yin Tang, Kai Kei Cheng, Raye Chen-Hua Yeow, Kai Keng Ang, Cuntai Guan, Jeong Hoon Lim, Brain-Computer Interface-based soft robotic glove rehabilitation for stroke, IEEE Trans. Biomed. Eng. (2020).

[13] Andrew B. Schwartz, X. Tracy Cui, Douglas J. Weber, Daniel W. Moran, Brain-controlled interfaces: movement restoration with neural prosthetics, Neuron (2006).

[14] Meel Velliste, Sagi Perel, M Chance Spalding, Andrew S Whitford, Andrew B Schwartz, Cortical control of a prosthetic arm for self-feeding, Nature (2008).

[15] Kapil D. Katyal, Matthew S. Johannes, Spencer Kellis, Tyson Aflalo, Christian Klaes, Timothy G. McGee, Matthew P. Para, Ying Shi, Brian Lee, Kelsie Pejsa, et al., A collaborative BCI approach to autonomous control of a prosthetic limb system, in: 2014 IEEE International Conference on Systems, Man, and Cybernetics, SMC, 2014.

[16] Sorin M. Grigorescu, Thorsten Lüth, Christos Fragkopoulos, Marco Cyriacks, Axel Gräser, A BCI-controlled robotic assistant for quadriplegic people in domestic and professional life, Robotica (2012).

[17] Sebastian Schröer, Ingo Killmann, Barbara Frank, Martin Völker, Lukas Fiederer, Tonio Ball, Wolfram Burgard, An autonomous robotic assistant for drinking, in: 2015 IEEE International Conference on Robotics and Automation, ICRA, 2015.

[18] Daniel Kuhner, Lukas D.J. Fiederer, Johannes Aldinger, Felix Burget, Martin Völker, Robin T. Schirrmeister, Chau Do, Joschka Boedecker, Bernhard Nebel, Tonio Ball, et al., Deep learning based BCI control of a robotic service assistant using intelligent goal formulation, 2018, BioRxiv.

[19] Gert Pfurtscheller, Event-related synchronization (ERS): an electrophysiological correlate of cortical areas at rest, Electroencephalogr. Clin. Neurophysiol. (1992).

[20] Christa Neuper, Michael Wörtz, Gert Pfurtscheller, ERD/ERS patterns reflecting sensorimotor activation and deactivation, Prog. Brain Res. (2006).

[21] Leonard J. Trejo, Roman Rosipal, Bryan Matthews, Brain-computer interfaces for 1-D and 2-D cursor control: designs using volitional control of the EEG spectrum or steady-state visual evoked potentials, IEEE Trans. Neural Syst. Rehabil. Eng. (2006).

[22] Pablo Martinez, Hovagim Bakardjian, Andrzej Cichocki, Fully online multicommand brain-computer interface with visual neurofeedback using SSVEP paradigm, Comput. Intell. Neurosci. (2007).

[23] Lawrence Ashley Farwell, Emanuel Donchin, Talking off the top of your head: toward a mental prosthesis utilizing event-related brain potentials, Electroencephalogr. Clin. Neurophysiol. (1988).

[24] Leigh R. Hochberg, Daniel Bacher, Beata Jarosiewicz, Nicolas Y. Masse, John D. Simeral, Joern Vogel, Sami Haddadin, Jie Liu, Sydney S. Cash, Patrick Van Der Smagt, et al., Reach and grasp by people with tetraplegia using a neurally controlled robotic arm, Nature (2012).

[25] Jennifer L. Collinger, Brian Wodlinger, John E. Downey, Wei Wang, Elizabeth C. Tyler-Kabara, Douglas J. Weber, Angus J.C. McMorland, Meel Velliste, Michael L. Boninger, Andrew B. Schwartz, High-performance neuroprosthetic control by an individual with tetraplegia, Lancet (2013).

[26] Michele Barsotti, D. Leonardis, C. Loconsole, Massimiliano Solazzi, E. Sotgiu, C. Procopio, C Chisari, M. Bergamasco, A. Frisoli, A full upper limb robotic exoskeleton for reaching and grasping rehabilitation triggered by MI-BCI, in: 2015 IEEE International Conference on Rehabilitation Robotics, ICORR, IEEE, 2015, pp. 49–54.

[27] Wenchang Zhang, Fuchun Sun, Jianhua Chen, Chuanqi Tan, Hang Wu, Weihua Su, An asynchronous Mi-based BCI for brain-actuated robot grasping control, in: 2017 International Conference on Computer Systems, Electronics and Control, ICCSEC, IEEE, 2017, pp. 893–898.

[28] Jeong-Hyun Cho, Ji-Hoon Jeong, Kyung-Hwan Shim, Dong-Joo Kim, Seong-Whan Lee, Classification of hand motions within EEG signals for non-invasive BCI-based robot hand control, in: 2018 IEEE International Conference on Systems, Man, and Cybernetics, SMC, IEEE, 2018, pp. 515–518.

[29] Rossella Spataro, Antonio Chella, Brendan Allison, Marcello Giardina, Rosario Sorbello, Salvatore Tramonte, Christoph Guger, Vincenzo La Bella, Reaching and grasping a glass of water by locked-in ALS patients through a BCI-controlled humanoid robot, Front. Hum. Neurosci. 11 (2017) 68.

[30] Filippo Arrichiello, Paolo Di Lillo, Daniele Di Vito, Gianluca Antonelli, Stefano Chiaverini, Assistive robot operated via P300-based brain computer interface, in: 2017 IEEE International Conference on Robotics and Automation, ICRA, IEEE, 2017, pp. 6032–6037.

[31] Batyrkhan Saduanov, Tohid Alizadeh, Jinung An, Berdakh Abibullaev, Trained by demonstration humanoid robot controlled via a BCI system for telepresence, in: 2018 6th International Conference on Brain-Computer Interface, BCI, IEEE, 2018, pp. 1–4.

[32] Jonathan Delijorge, Omar Mendoza-Montoya, Jose L. Gordillo, Ricardo Caraza, Hector R. Martinez, Javier M. Antelis, Evaluation of a P300-based brain-machine interface for a robotic hand-orthosis control, Front. Neurosci. 14 (2020).

[33] Arnab Rakshit, Amit Konar, Atulya K. Nagar, A hybrid brain-computer interface for closed-loop position control of a robot arm, IEEE/CAA J. Autom. Sin. 7 (5) (2020) 1344–1360.

[34] Saugat Bhattacharyya, Amit Konar, D.N. Tibarewala, Motor imagery and error related potential induced position control of a robotic arm, IEEE/CAA J. Autom. Sin. 4 (4) (2017) 639–650.

[35] Jingsheng Tang, Zongtan Zhou, A shared-control based BCI system: For a robotic arm control, in: 2017 First International Conference on Electronics Instrumentation & Information Systems, EIIS, IEEE, 2017, pp. 1–5.

[36] Yang Xu, Cheng Ding, Xiaokang Shu, Kai Gui, Yulia Bezsudnova, Xinjun Sheng, Dingguo Zhang, Shared control of a robotic arm using non-invasive brain–computer interface and computer vision guidance, Robot. Auton. Syst. 115 (2019) 121–129.

[37] Yang Xu, Heng Zhang, Linfeng Cao, Xiaokang Shu, Dingguo Zhang, A shared control strategy for reach and grasp of multiple objects using robot vision and noninvasive brain-computer interface, IEEE Trans. Autom. Sci. Eng. (2020).

[38] Yiliang Liu, Wenbin Su, Zhijun Li, Guangming Shi, Xiaoli Chu, Yu Kang, Weiwei Shang, Motor-imagery-based teleoperation of a dual-arm robot performing manipulation tasks, IEEE Trans. Cogn. Dev. Syst. 11 (3) (2018) 414–424.

[39] Zhichuan Tang, Lingtao Zhang, Xin Chen, Jichen Ying, Xinyang Wang, Hang Wang, Wearable supernumerary robotic limb system using a hybrid control approach based on motor imagery and object detection, IEEE Trans. Neural Syst. Rehabil. Eng. 30 (2022) 1298–1309.

[40] Hong Zeng, Yitao Shen, Xuhui Hu, Aiguo Song, Baoguo Xu, Huijun Li, Yanxin Wang, Pengcheng Wen, Semi-autonomous robotic arm reaching with hybrid gaze–brain machine interface, Front. Neurorobotics 13 (2020) 111.

[41] Xingchao Wang, Xiaopeng Huang, Yi Lin, Liguang Zhou, Zhenglong Sun, Yangsheng Xu, Design of an SSVEP-based BCI stimuli system for attention-based robot navigation in robotic telepresence, in: 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, IEEE, 2021, pp. 4126–4131.

[42] Wenchang Zhang, Fuchun Sun, Chunfang Liu, Weihua Su, Chuanqi Tan, Shaobo Liu, A hybrid EEG-based BCI for robot grasp controlling, in: 2017 IEEE International Conference on Systems, Man, and Cybernetics, SMC, IEEE, 2017, pp. 3278–3283.

[43] Paolo Di Lillo, Filippo Arrichiello, Daniele Di Vito, Gianluca Antonelli, BCI-controlled assistive manipulator: developed architecture and experimental results, IEEE Trans. Cogn. Dev. Syst. (2020).

[44] Xiaoqian Mao, Wei Li, Chengwei Lei, Jing Jin, Feng Duan, Sherry Chen, A brain–robot interaction system by fusing human and machine intelligence, IEEE Trans. Neural Syst. Rehabil. Eng. 27 (3) (2019) 533–542, http://dx.doi.org/10.1109/TNSRE.2019.2897323.

[45] Iason Batzianoulis, Fumiaki Iwane, Shupeng Wei, Carolina Gaspar Pinto Ramos Correia, Ricardo Chavarriaga, José del R. Millán, Aude Billard, Customizing skills for assistive robotic manipulators, an inverse reinforcement learning approach with error-related potentials, Commun. Biol. 4 (1) (2021) 1–14.

[46] Fred Paas, Juhani E. Tuovinen, Huib Tabbers, Pascal W.M. Van Gerven, Cognitive load measurement as a means to advance cognitive load theory, Educ. Psychol. 38 (1) (2003) 63–71.

[47] Pavlo Antonenko, Fred Paas, Roland Grabner, Tamara Van Gog, Using electroencephalography to measure cognitive load, Educ. Psychol. Rev. 22 (4) (2010) 425–438.

[48] Naveen Kumar, Jyoti Kumar, Measurement of cognitive load in HCI systems using EEG power spectrum: an experimental study, Procedia Comput. Sci. 84 (2016) 70–78.

[49] Sandra G. Hart, Lowell E. Staveland, Development of NASA-TLX (Task Load Index): Results of empirical and theoretical research, in: Advances in Psychology, Vol. 52, Elsevier, 1988, pp. 139–183.

[50] Aniana Cruz, Gabriel Pires, Ana Lopes, Carlos Carona, Urbano J. Nunes, A self-paced BCI with a collaborative controller for highly reliable wheelchair driving: Experimental tests with physically disabled individuals, IEEE Trans. Hum.-Mach. Syst. 51 (2) (2021) 109–119.

[51] Francisco Velasco-Álvarez, Álvaro Fernández-Rodríguez, Ricardo Ron-Angevin, Brain-computer interface (BCI)-generated speech to control domotic devices, Neurocomputing 509 (2022) 121–136.

[52] Kaiming He, Georgia Gkioxari, Piotr Dollár, Ross Girshick, Mask r-cnn, in: Proceedings of the IEEE International Conference on Computer Vision, 2017.

[53] Yun Jiang, Stephen Moseson, Ashutosh Saxena, Efficient grasping from rgbd images: Learning using a new rectangle representation, in: 2011 IEEE International Conference on Robotics and Automation, 2011.

[54] Ian Lenz, Honglak Lee, Ashutosh Saxena, Deep learning for detecting robotic grasps, Int. J. Robot. Res. (2015).

[55] Ashutosh Saxena, Justin Driemeyer, Andrew Y. Ng, Robotic grasping of novel objects using vision, Int. J. Robot. Res. (2008).

[56] Matei Ciocarlie, Kaijen Hsiao, Edward Gil Jones, Sachin Chitta, Radu Bogdan Rusu, Ioan A. Şucan, Towards reliable grasping and manipulation in household environments, in: Experimental Robotics, Springer, 2014.

[57] Joseph Redmon, Anelia Angelova, Real-time grasp detection using convolutional neural networks, in: 2015 IEEE International Conference on Robotics and Automation, ICRA, 2015.

[58] Sulabh Kumra, Christopher Kanan, Robotic grasp detection using deep convolutional neural networks, in: 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, 2017.

[59] Xinwen Zhou, Xuguang Lan, Hanbo Zhang, Zhiqiang Tian, Yang Zhang, Narming Zheng, Fully convolutional grasp detection network with oriented anchor box, in: 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, 2018.

[60] Fu-Jen Chu, Ruinian Xu, Patricio A. Vela, Real-world multiobject, multigrasp detection, IEEE Robot. Autom. Lett. (2018).

[61] Fabien Lotte, Cuntai Guan, Regularizing common spatial patterns to improve BCI designs: unified theory and new algorithms, IEEE Trans. Biomed. Eng. (2010).

[62] Svante Wold, Kim Esbensen, Paul Geladi, Principal component analysis, Chemometr. Intell. Lab. Syst. (1987) 37–52.

[63] H. Zhang, X. Lan, S. Bai, X. Zhou, Z. Tian, N. Zheng, ROI-based robotic grasp detection for object overlapping scenes, in: 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, 2019.

[64] H. Zhang, X. Lan, S. Bai, L. Wan, C. Yang, N. Zheng, A multi-task convolutional neural network for autonomous robotic grasping in object stacking scenes, in: 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, 2019.

[65] Dongwon Park, Yonghyeok Seo, Dongju Shin, Jaesik Choi, Se Young Chun, A single multi-task deep neural network with post-processing for object detection with reasoning and robotic grasp detection, in: 2020 IEEE International Conference on Robotics and Automation, ICRA, 2020.

[66] Ross Girshick, Fast r-cnn, in: Proceedings of the IEEE International Conference on Computer Vision, 2015.

[67] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016.

[68] Shaoqing Ren, Kaiming He, Ross Girshick, Jian Sun, Faster r-cnn: Towards real-time object detection with region proposal networks, in: Advances in Neural Information Processing Systems, 2015.

[69] World Medical Association, et al., World Medical Association Declaration of Helsinki. Ethical principles for medical research involving human subjects, Bull. World Health Organ. 79 (4) (2001) 373.

[70] Lianghua He, Die Hu, Meng Wan, Ying Wen, Karen M. Von Deneen, MengChu Zhou, Common Bayesian network for classification of EEG-based multiclass motor imagery BCI, IEEE Trans. Syst. Man Cybern. 46 (6) (2015) 843–854.

[71] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, Alexander C. Berg, Ssd: Single shot multibox detector, in: European Conference on Computer Vision, 2016.

[72] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al., Imagenet large scale visual recognition challenge, Int. J. Comput. Vis. (2015).

[73] Jason Wang, Luis Perez, The effectiveness of data augmentation in image classification using deep learning, Convolutional Neural Netw. Vis. Recognit. (2017).

[74] Sebastien C. Wong, Adam Gatt, Victor Stamatescu, Mark D. McDonnell, Understanding data augmentation for classification: when to warp? in: 2016 International Conference on Digital Image Computing: Techniques and Applications, DICTA, IEEE, 2016.

[75] Connor Shorten, Taghi M. Khoshgoftaar, A survey on image data augmentation for deep learning, J. Big Data (2019).

[76] Fei Pan, Baoying Wang, Xin Hu, William Perrizo, Comprehensive vertical sample-based KNN/LSVM classification for gene expression analysis, J. Biomed. Inform. (2004).

[77] Ujwala Ravale, Nilesh Marathe, Puja Padiya, Feature selection based hybrid anomaly intrusion detection system using K means and RBF kernel function, Procedia Comput. Sci. (2015).

[78] Anisha Halder, Amit Konar, Rajshree Mandal, Aruna Chakraborty, Pavel Bhowmik, Nikhil R. Pal, Atulya K. Nagar, General and interval type-2 fuzzy face-space approach to emotion recognition, IEEE Trans. Syst. Man Cybern. (2013).

[79] Anuradha Saha, Amit Konar, Atulya K. Nagar, EEG analysis for cognitive failure detection in driving using type-2 fuzzy classifiers, IEEE Trans. Emerg. Top. Comput. Intell. (2017).

[80] Vu N.P. Dao, V.R. Vemuri, A performance comparison of different back propagation neural networks methods in computer network intrusion detection, Differ. Equ. Dyn. Syst. (2002).

[81] Alex Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton, Imagenet classification with deep convolutional neural networks, in: Advances in Neural Information Processing Systems, 2012.

[82] Daniel Griffin, Jae Lim, Signal estimation from modified short-time Fourier transform, IEEE Trans. Acoust. Speech Signal Process. (1984).

[83] Arne Jensen, Anders la Cour-Harbo, Ripples in Mathematics: The Discrete Wavelet Transform, Springer Science & Business Media, 2001.

[84] Garett D. Johnson, Dean J. Krusienski, Ensemble SWLDA classifiers for the P300 speller, in: International Conference on Human-Computer Interaction, Springer, 2009, pp. 551–557.

[85] Aapo Hyvärinen, Erkki Oja, Independent component analysis: algorithms and applications, Neural Netw. (2000) 411–430.

[86] Hanchuan Peng, Fuhui Long, Chris Ding, Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy, IEEE Trans. Pattern Anal. Mach. Intell. 27 (8) (2005) 1226–1238.

[87] Hanbo Zhang, Xuguang Lan, Xinwen Zhou, Zhiqiang Tian, Yang Zhang, Nanning Zheng, Visual manipulation relationship network for autonomous robotics, in: 2018 IEEE-RAS 18th International Conference on Humanoid Robots (Humanoids), 2018.