

הקדמה-

ממשק מוח-מחשב (brain-computer interface, BCI) הוא ממשק ישיר בין המוח למערכת מלאכותית שמטרתו לשלוט על הפעלת מכשיר (רובוט, כיסא גלגלים וכו') על ידי שימוש בפעילות החשמלית המוחית וללא שימוש בעצבים או שרירים היקפיים. מטרת תרגיל זה, הייתה לנתח נתוני הקלטה מ-EEG (רישום פעילות חשמלית מוחית) של נבדק בעת שהוא מדמין פעילות מוטורית (motor imagery, MI) - הזזת יד ימין או יד שמאל, ננסה לחלץ תכונות (features) שמשמעותן מידע מאפיין של הנתונים הגולמיים כמו גודל, עוצמה, צורה וכדומה. בניסוי בוצעו 160 ניסיונות (trials) אשר הוקלטו מ-2 אלקטרודות (ערצים C3, C4). עבור 128 ניסיונות קיבלנו את התיוגים המתאימים. בעזרת תיוגים אלו נוכל לבצע אימון מסווג (classifier). אימון זה יתבצע בעזרת שיטת Linear Discriminant Analysis (LDA), שיטה למציאת הפרדה ליניארית בין שתי מחלקות. תהליך הלמידה והבדיקה יכול להתבצע במגוון דרכים שונות, בפרויקט שלנו נבצע את התהליך בעזרת שיטת k-fold cross validation מתוך שיקולים של עלות שימוש בנתונים אל מול יעילות. שיטה זו מחלקת את הנתונים לא חלקים אקראיים ובכל פעם קטע אחד משמש כקבוצת מבחן ושאר k-1 החלקים משמשים כקבוצת אימון לסיווג. תהליך זה יעזור לנו להבין אם אכן ביצענו הפרדה נכונה למחלקות השונות. לאחר מכן נוכל לבחון את תוצאותינו על 32 ניסיונות שלא ביצענו עליהם למידה ותיוגם אינו ידוע (הסבר מפורט בפרק השיטות). לאחר תהליך זה נרצה להבין אם הצלחנו לסווג בין מחלקה שמאל לימין.

אנו משערים כי נוכל למצוא תכונות אשר יעזרו לנו לסווג את המחלקות השונות ולהבדילם זו מזו.

שיטות-

בכדי להצליח בתהליך הסיווג למחלקות השונות, פעלנו במספר דרכי פעולה אשר יעזרו לנו בתהליך:

- ניהול הנתונים וסינונם בהתאם לדרישות הפרויקט (פונקציית **dataHandle**).
- ויזואליזציה של המידע הגולמי (**visualization**)
- חישוב והצגת **Power spectrum** לנתונים הגולמיים
- חישוב והצגת **Spectrogram** לנתונים הגולמיים
- הוצאת תכונות (**Features**), הצגתם באופן ויזואלי בהיסטוגרמה וביצוע תהליך **Principal component algorithm (Pca)**.
- סיווג (**Classification**) – אימון ובדיקה.

כעת נרחיב על השיטות והפונקציות השונות.

פונקציית dataHandle -

את המידע עבור המחלקות (ימין ושמאל) נקבל כקלט. נסנן מידע עבורם בכל הערוצים (C3,C4) ונחזיר כפלט את קצב דגימת הנתונים ואת cell הנתונים שלנו בשם data שבכל שורה מידע עבור מחלקה מסוימת בכל ערוציה. בקוד נרוץ על מספר המחלקות, נוציא אינדקס המתאים למחלקה הנוכחית מהרשימה, עבור מחלקה זו נמצא את האינדקסים בהם הנתונים אכן מסווגים למחלקה זו ונחלץ את כלל ערוצי המחלקה ממטריצת הקלטת ה-EEG.

```
function [outputData,fs]= dataHandle(classes,nClasses)
for i=1:nClasses
    % filter data only from input classes (left and right)
    currInd=strcmpi(classes(i),data.attributename); % row will be 1 if class exists in
data, if not row is 0.
    trials=data.attribute(currInd,:) == 1; %trials locataion who belongs to current
class
    outputData{i}=EEG_mat(trials,,:); % data containing trials from class
end
end
```

:Visualization

בכדי לקבל השערה למאפיינים המפרידים של המחלקות, נרצה להתבונן בנתונים הגולמיים בצורה ויזואלית. ראשית נציג עבור כל מחלקה (ימין ושמאל) את הנתונים באופן ויזואלי. נרוץ בלולאה על מספר המחלקות, בכל צעד נמפה 20 ניסיונות (trials) רנדומליים מתוך הנתונים בעזרת קריאה לפונקציה visualizeClass, בה נגדיר וקטור אינדקסים רנדומלי (פונקציית randperm) שבעזרתו נוכל להצביע על trial אקראי בנתונים המתאימים למחלקה הנוכחית, נציג בתת גרף את המתח ביחידות μV עבור trial זה, עבור 2 הערוצים (C3 וC4) על פני משך זמן ההקלטה (6 שניות).

```
function
visualizeClass(time_vec,data,fs,plotScales,classes,channels,loc,maxFreq,nRandTrials)
...
randTrials=randperm(size(data,1),nRandTrials); % get random indexes representing
trials
for i=1:nRandTrials
    subplot (rows,cols,i);
    %plots all channels (C3 and C4) from current trial in one subplot
    hold on;
    for chan=1:size(channels,2)
        plot(time_vec/fs,data(randTrials(i),:,chan));
    end
    hold off;
```

Power spectrum:

נחשוב על הסיגנל שלנו כפונקציה של עוצמת הפעילות החשמלית כתלות בזמן ונפעיל עליה את התמרת פוריה ונקבל את עוצמת הגלים (התדרים) המרכיבים את הסיגנל. העוצמה כפונקציה של התדר נקראת **Power spectrum (PS)**. בכך נוכל נפרק את הסיגנל למרכיביו.

נחשב עבור כל מחלקה את ה-PS. נרוץ בלולאה על מספר המחלקות ובכל צעד נקרא לפונקציה `getPwelch` המחשבת את ה-PS עבור כל הערוצים הנתונים בלולאה על מספר הערוצים, ומכניסה אותם ל-cell כאשר כל שורה מייצגת מטריצה של ערוץ. ה-PS הנוכחי מחושב בקריאה לפונקציה המובנית `pwelch` המקבלת מטריצת נתונים, וקטור תדרים וקצב דגימה (`f,fs,data`) ומחשבת עוצמה בעזרת חלוקה לחלונות באורך 2 שניות עם חפיפה של שניה (`pwWindow, pwOverlap`). כמו כן הפונקציה `getPwelch` מקבלת כקלט את זמן התחלת חישוב העוצמה (`startInd`), בחרנו בשנייה 4 לאחר ניסוי וטעיה ושיערנו כי זהו הזמן העיקרי בו מתבצעת `motor imagery` ובכך נקבל תוצאה המבדילה בין התדרים באופן מיטבי. לבסוף נכניס את הפלט לשורה המתאימה למחלקה ב-cell בשם `pwelches`. כלומר נקבל cell לכלל העוצמות המכיל בכל שורה cell המייצג מחלקה מסוימת. כל מחלקה מכילה בכל שורה מטריצה של נתוני הערוץ. (החלוקה נובעת מנוחות הכנסת cell שלם המייצג מחלקה לפונקציות הבאות). באופן טכני:

`pwelches cell= 2 rows, each row is a cell with 2 matrices size- (frequencies x trials)`

תצוגה גרפית של ה- **Power spectrum (PS)** -

נציג כל PS בתת גרף משלו בעזרת קריאה לפונקציה `plotPwelch`, המדפיסה כל ערוץ ממחלקה מסוימת בתת גרף משלו. (תצוגה ויזואלית לגבי הערוצים והמחלקות). כך נוכל להסיק את תחום התדרים בהם העוצמה מוגברת עבור כל תת גרף.

בנוסף בכדי להבין את הנתונים בצורה מעמיקה יותר, ניצור גרף עבור כל ערוץ. נרוץ בלולאה על מספר הערוצים, בכל צעד נקרא לפונקציה `plotChannelPwelch` אשר מציגה על גרף את ה-PS של כלל המחלקות (שמאל וימין) על ידי ריצה בלולאה על מספר המחלקות והוספתם לגרף הערוץ הנוכחי. נעשה זאת על מנת שנוכל לראות הבדל ויזואלי בין עוצמות ה-PS של המחלקות על פני רצף התדרים השונים ולמצוא רצועות תדרים משמעותיות להוצאת תכונות עבורן.

Spectrogram:

ספקטוגרמה היא ייצוג ויזואלי של ספקטרום תדרים על פני רצף זמן. בכדי למצוא רצועות תדרים אינפורמטיביות, ניצור ספקטוגרמה עבור כל מחלקה וערוציה. נרוץ בלולאה על מספר המחלקות ועבור כל מחלקה נוסיף את הספקטוגרמה של כל ערוציה ל-cell בשם `specs` באינדקס המתאים לה בעזרת קריאה לפונקציה `getSpectrogram`. הפונקציה מחשבת את הספקטוגרמה לכל trial (לולאה על `nTrails`) בעזרת קריאה לפונקציה `spectrogram`. הפונקציה נעזרת בחלוקה לחלונות בגודל שנייה וחפיפה של 0.8 שניות בין החלונות. לאחר מכן את מטריצת הספקטוגרמה בתא הנוכחי נמיר לדציבלים (הכפלה ב- $10 \cdot \log$) על מנת לאבחן שינויים קטנים בצורה טובה יותר. כמו כן נמצע על

פני trials לקבלת ספקטוגרמה ממוצעת (נוח יותר מאשר לבדוק 128 ספקטוגרמות שונות). לבסוף נקבל cell עבורו כל שורה היא מטריצת הספקטוגרמה לערוץ במחלקה. בשונה מחישוב הPS, נחשב את הספקטוגרמה על פני משך כל זמן הניסוי (6 שניות) על מנת להבחין בהבדלים בעוצמות על פני הזמן ולאתר נקודות זמן בהן קיים תחום תדרים בולט. קוד הפונקציה:

```
function [specs,currInd]=getSpectrogram(specs,data,currInd,fs,nTrials,nChannels)
...
for i=1:nChannels
    currSpec=i+currInd; % each call needs to insert in correct location in specs
    specs{currSpec}=zeros(size(s,1),size(s,2),fs/2); % allocate current data
    for trial=1:(nTrials/2)
        %get spectrogram per trial using matlab function
        [~,~,specs{currSpec}(:, :, trial)]=spectrogram(data(trial, :, i),sg_window,sg_overlap, [],
        fs, 'power');
    end
    % convert units to dB and get average for all trials
    specs{currSpec}=mean(10*log10(specs{currSpec}),3);
end
currInd=currInd+nChannels; % update index after all channels inserted
end
```

תצוגה גרפית של הספקטוגרמות-

לאחר שקיבלנו את הספקטוגרמות השונות, נציג אותם בתת גרף לכל מחלקה וערוץ בעזרת קריאה לפונקציה plotSpectrogram אשר מקבלת את cell הספקטוגרמות ומציגה כל שורה בקריאה לפונקציה המובנית imagesc על פני רצף הזמן (ציר x) והתדרים (ציר y) כאשר כל ריבוע בשנייה מסוימת מייצג את הPower spectrum בדציבלים בתדר המתאים לה.

כמו כן כדי להבין טוב יותר את הנתונים נרוץ בלולאה על מספר הערוצים ונציג את ההפרשים בין המחלקות בכל ערוץ, על ידי קריאה לפונקציה plotSpecsDiff. הפונקציה נעזרת בדומה להצגת הספקטוגרמה הקודמת בפונקציה המובנית imagesc אך הפעם מציגה את הפרשי הספקטוגרמות (מיקומם ניתן כקלט) על מנת לזהות בבירור רצועות תדרים בהם ההפרש מובחן ויזואלית.

:Features

לאחר תהליך הוויזואליזציה בדרכים השונות לנתונים הגולמיים, נוכל להוציא מהמידע תכונות (Features) אשר יעזרו לנו בתהליך הסיווג (בפרק המבוא הסברנו מהי תכונות). בצירוף ידע נצבר מהקורס ומהספרות המקצועית, נוכל לאבחן את הגורמים עבורם מתבצעת ההפרדה הטובה ביותר למחלקות השונות (שמאל וימין). יש לציין שביצענו תהליך של ניסוי ותהייה הנעשה באופן ידני ועל כן נפרט בעמוד הבא את התכונות השונות:

את רוב התכונות (features) ניצור בעזרת פונקציית bandpower- חישוב האנרגיה, כלומר השטח תחת עוצמה בטווח תדרים מסוים, בזמן מסוים. בחרנו רצועות תדרים שונות אשר נראו רלוונטיות ויזואלית, כולן בתחום זמנים שבין 4-6 שניות.

תכונות relative log power הנלמדו במטלות קודמות הן תכונות אשר מחשבות את העוצמה היחסית של תחום תדרים מסוים ביחס לעוצמה הכוללת. בנוסף מבצעות המרה לא ליניארית ($10 \cdot \log_{10}$) לנתונים, המרה אשר מדגישה הבדלים קטנים בין העוצמות. בחרנו להתמקד בתחום 15-18 Hz ביחס לתחום הכולל 0-40 Hz בזמנים 4-6 שניות שבהם הנבדק פעיל יותר. תכונות root power הנלמדו במטלות קודמות מחזירות את השורש של עוצמה בתחום מסוים- פרופורציונלי לסטיית התקן של הסיגנל המקורי. בחרנו בתחום 14-18 Hz בשניות 4-6 מכיוון שהוא פעיל מאוד בשתי הערוצים במחלקה ימין.

כדי לאסוף את התכונות נאתחל מטריצת תכונות בשם features בגודל כמות התכונות המקסימלית שנרצה לאסוף כפול כמות הטריילרים (9×128). את התכונות ניצור בעזרת קריאה לפונקציה extractFeatures, תחילה על נתוני האימון ולבסוף באופן זהה על נתוני הבדיקה testData. ניצור רשימת פרמטרים- תחומי תדר ורשימת ערוצים בהתאמה לסוג התכונות ומיקומה בתהליך ההוצאה. נרוץ על כל סוג תכונות בלולאה על מספר התכונות שנרצה להוציא מסוג זה. נחשב את העוצמה בתחום התדר המתאים ($\text{featsPrmt}(i).\text{freqs}(i)$), לערוץ המתאים בזמן המתאים עבור כל trials של הקלטת הנתונים ($\text{data}(:, \text{sInd:end}, \text{featsPrmt}(i).\text{chan}(i))$) לפי קצב דגימה ידוע (fs) בעזרת פונקציית bandpower ונוסיף את התכונות למטריצת פלט. נציג לדוגמה חלק מהקוד, עבור חישוב root power:

```
function
[features, iFeat, types]=extractFeatures(features, nFeatures, data, fs, featPrmt, sInd, iFeat)
...
% root power features
for i=1:size(featsPrmt(3).chan,2)
    features(iFeat,:)=sqrt(bandpower(data(:,sInd:end,featsPrmt(3).chan(i))',fs,
    featsPrmt(3).freqs{i}));
    types{iFeat}="Root Power";
    iFeat=iFeat+1;
end
end
```

פונקציית plotFeaturesHistogram -

בעזרת קריאה לפונקציה זו נציג בגרף נתונים משותפים משתי המחלקות (ימין ושמאל), כל תכונת feature) תוצג בתת גרף על ידי היסטוגרמה (histogram פונקציית מובנית בתוכנה). נרוץ פעמיים את הפונקציה histogram, בכל פעם עבור ניסיונות (trials) השייכים למחלקה מסוימת – שמאל וימין. פונקציה זו תחלק את המידע הנוכחי למספר מקטעים, בכל מקטע נוכחי תסכם כמות הפעמים שהתכונת הופיעה. עם כלי נבחן בצורה חזותית האם התכונת אכן מפרידה בין המחלקות השונות.

-(PCA) Principal Component Algorithm

לאחר תהליך הוצאת התכונות, נבצע תהליך של ניתוח גורמים ראשיים - Principal component algorithm (Pca). זוהי שיטה סטטיסטית של הפחתת ממדים אשר מתמירה את הנתונים למערכת קואורדינטות אורתוגונלית ומוצאת את הרכיבים בעלי השונות הגבוהה ביותר. שיטה זו תעזור לנו להבין האם ניתן ליצור חיץ בין המחלקות השונות, בעזרת הפחתת ממדי התכונות, הדבר יתבצע בעזרת קריאה לפונקציה `plotPca`. פונקציה זו תחשב את הווקטורים העצמיים המתאימים לערכים העצמיים הגדולים ביותר נסמנם בשם `eigs` (נחשב בעזרת פונקציה מובנית `Pca`) בעזרתה נציג את התכונות המומרות שהן הרכיבים המשמעותיים ביותר על ידי הכפלת הווקטורים העצמיים `eigs` במטריצת התכונות לאחר הפחתת ממוצעי התכונות (`reducedF`) כעת נוכל להציג את תוצאות התהליך בחלוקה למחלקות שמאל וימין (הצגה דו ממדית והצגה תלת ממדית).

:Classification

אימון:

בכדי לדעת האם התכונות שלנו אכן מחלקות את המידע לשמאל וימין בצורה נכונה, נרצה לאמן בעצמינו את הסיווג. את תהליך המיון והסיווג נבצע בעזרת שיטת `K fold cross validation`. המוזכרת בפרק המבוא. שיטה זו תתבצע בעזרת קריאה לפונקציה `classifyTrain`. בשיטה זו נחלק רנדומלית למקטעים (`folds`) את המידע השמור במטריצת התכונות שלנו- ניצור וקטור עם ערכים מ-1 עד מספר הניסיונות (128) רנדומלית עם פונקציית `randperm` ונחזיר את השארית של כל איבר בחלוקה לא ובכך נקבל 128 אינדקסים שכל איבר מייצג קבוצה ייחודית בין 1 ל- k). בחרנו ב- $k=10$ על מנת לאזן בין אחוזי דיוק גבוהים וסטיית תקן נמוכה ככל הניתן- גודל קבוצה הוא מספר הניסיונות (`trials`) לחלק למספר הקבוצות `K`. כעת נרוץ בלולאה על מספר הקבוצות. בכל ריצה קבוצה אחת תחשב לקבוצת המבחן בעוד שאר הקבוצות (`k-1`) יישמשו כקבוצות האימון. את התגיות המתאימות (`labels`) הגדרנו מראש מתוך הנתונים שקיבלנו ולכן נדע מראש לסווג נכונה את כל קבוצות האימון בכל ריצה. כעת בעזרת פונקציה המוגדרת בתוכנה מטלאב, הנקראת `classify` נכניס את הנתונים בחלוקה לקבוצת מבחן, קבוצת אימון, וקבוצת סוגי הפרדה (וקטור התיוגים של המחלקות ימין ושמאל). הפונקציה תחשב בעזרת שיטת סיווג ליניארית- `Linear discriminant analysis` (LDA) את הסיווג של כל קבוצת מבחן, LDA מנסה למצוא הפרדה ליניארית של הנתונים ל-2 מחלקות.

בכל ריצה על קבוצה כלשהי, נשמור בווקטור את אחוזי הדיוק של קבוצות האימון וקבוצת המבחן (`test and validation sets`) בכל ריצה. הטעות של קבוצת האימון מוחזרת כפלט של פונקציית `classify` ולכן בקלות נחשב את הדיוק (`1-err`). את דיוק קבוצת הבדיקה נחשב בעזרת ממוצע התוצאות שסווגו נכונה (סכימת כמות התוצאות הנכונות לחלק לגודל קבוצת המבחן הנוכחית). לאחר הלולאה נמצע על פני כמות הקבוצות ונוציא סטיות תקן לשני סוגי הדיוקים (בעזרת פונקציות מובנות `mean, std`). בעמוד הבא מצורף קוד הפונקציה `classifyTrain`:

```

function [acc,sd] =classifyTrain(k,whichFeats,features,nTrials,labels)
folds= mod(randperm(nTrials),k)+1; % randomly choose folds
for i=1:k
    % define validation (test) set, training set and group set
    sample=features(whichFeats,folds==i);
    training=features(whichFeats,folds~=i);
    group=labels(folds~=i);
    % classify, get back classification result and train error
    [class(folds==i),trainErr]=classify(sample',training',group','linear');
    trainAcc(i)=1-trainErr; % train accuracy
    groupSample = labels(folds==i); % labels for validation set
    validationAcc(i) = mean( class(folds==i) == groupSample' ); % validation (test)
end
% compute total accuracy and std (validation & training)
acc=[mean(trainAcc),mean(validationAcc)];
sd=[std(trainAcc),std(validationAcc)];

```

בדיקה:

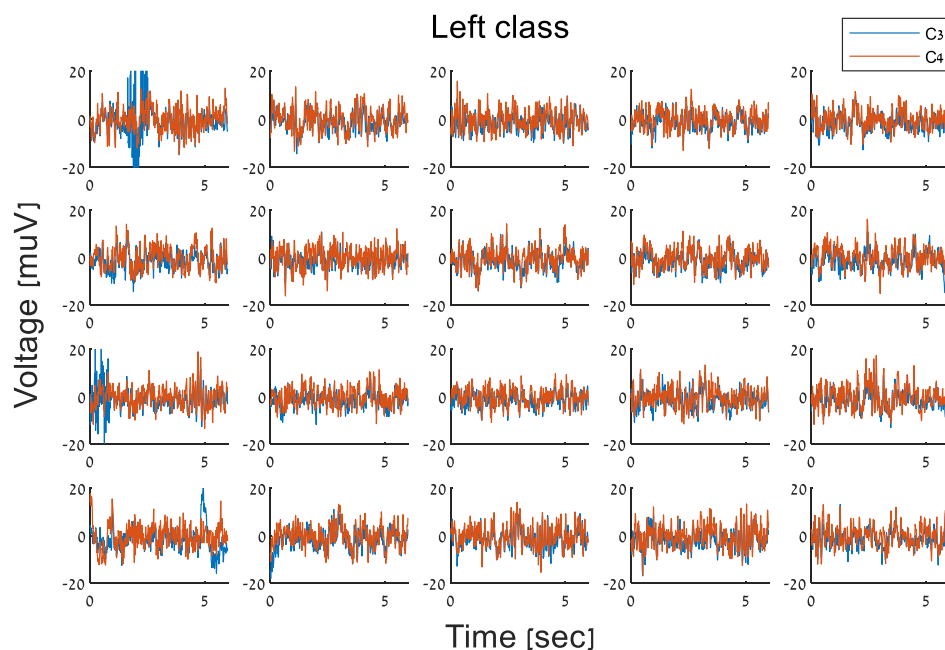
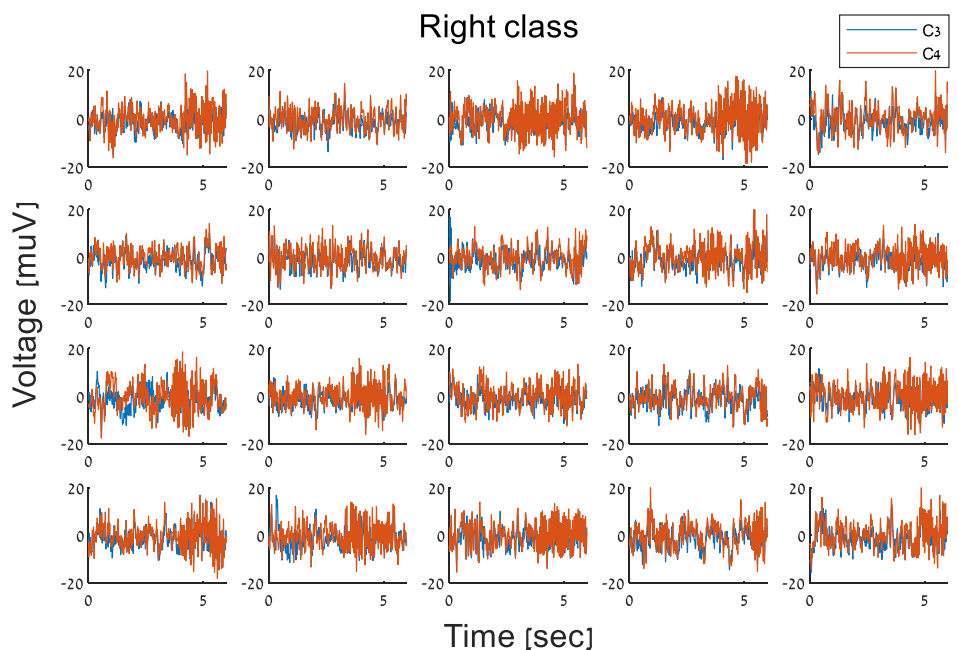
לאחר שאימנו מסווג, נרצה לבחון את יכולותיו, לשם כך שמרנו בצד 32 ניסיונות (trials) לבדיקה שעליהם לא התאמנו (תיגום ידוע רק לבוחנים חיצוניים ורק לאחר התהליך נדע את תוצאותינו). תהליך זה מאפשר לנו להימנע מoverfitting שהיא טעות הכללה הנובעת מלמידה פרטנית דקדקנית של נתונים ספציפיים אשר תפחית את הסיכוי ללמוד מאפיינים כללים אשר יפרידו בהצלחה בעתיד בין הגורמים השונים של נתונים שונים.

את אותם התכונות שאספנו במהלך הפרויקט נחשב כעת על נתוני הבדיקה השמורים (נשמור במטריצת testFeatures). בעזרת קריאה לפונקציה classify, נבצע סיווג ליניארי- קבוצת האימון הינה מטריצת התכונות (features) של האימון, קבוצת המבחן הינה מטריצת התכונות של הבדיקה testFeatures ותיוגי האימון הם קבוצת התגיות labels, נקבל סיווג לנתונים שלא ביצענו עליהם למידה.

תוצאות:

visualization:

מהצגת הניסיונות (trials) האקראיים ניתן להבחין באופן כללי בין דמיון דפוסי עלייה וירידה במתח בין הערוצים לאורך משך הניסוי, כאשר מתח בערוץ אחד עולה\ יורד, הערוץ השני מתנהג בצורה דומה (מלבד trials רועשים בודדים). קיימים הבדלים בעוצמת המתח בין הערוצים השונים בשתי המחלקות, עבור שתי המחלקות עוצמת המתח של ערוץ C4 חזקה יותר מעוצמת מתח ערוץ C3 (מלבד trials אקראיים רועשים). ניתן לראות את הזמנים בהם הנבדק פעיל יותר- עבור מחלקה ימין רוב הtrials מעידים על מתח גבוה יותר בטווח 3-6 שניות. עבור מחלקה שמאל עוצמת המתח נעה בתחומים דומים החל מהשנייה הראשונה. להלן תוצאות המחלקות:



תוצאות הפונקציה plotPwelch –

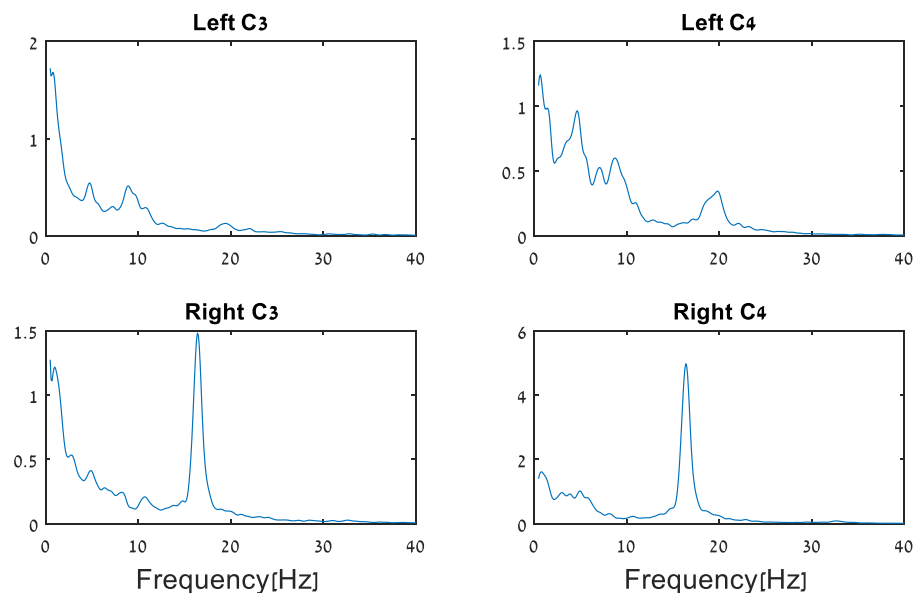
בהצגת כל Power Spectrum נוכל להבחין מהם התדרים בהם העוצמה חזקה עבור כל מחלקה וערוץ, להלן טבלה המסכמת את רצועות התדרים בהם התרחשה עלייה מקומית בעוצמה:

| Right C4 | Left C4 | Right C3 | Left C3 | Power spectrum |
|----------|---|----------|-----------------|--------------------------------------|
| [15-18] | [2.2-5], [6.3-7.8] [7.8-10], [17.5-21] | [15-18] | [4-5.5], [8-10] | רצועות תדרים (Hz) עם עוצמה מוגברת |

מנתונים אלו ניתן לראות כי עבור מחלקה ימין בשני הערוצים העוצמה מוגברת בתחום זהה.

עבור מחלקה שמאל, לכל ערוץ תחומים שונים בהם העוצמה חזקה.

Power Spectrums

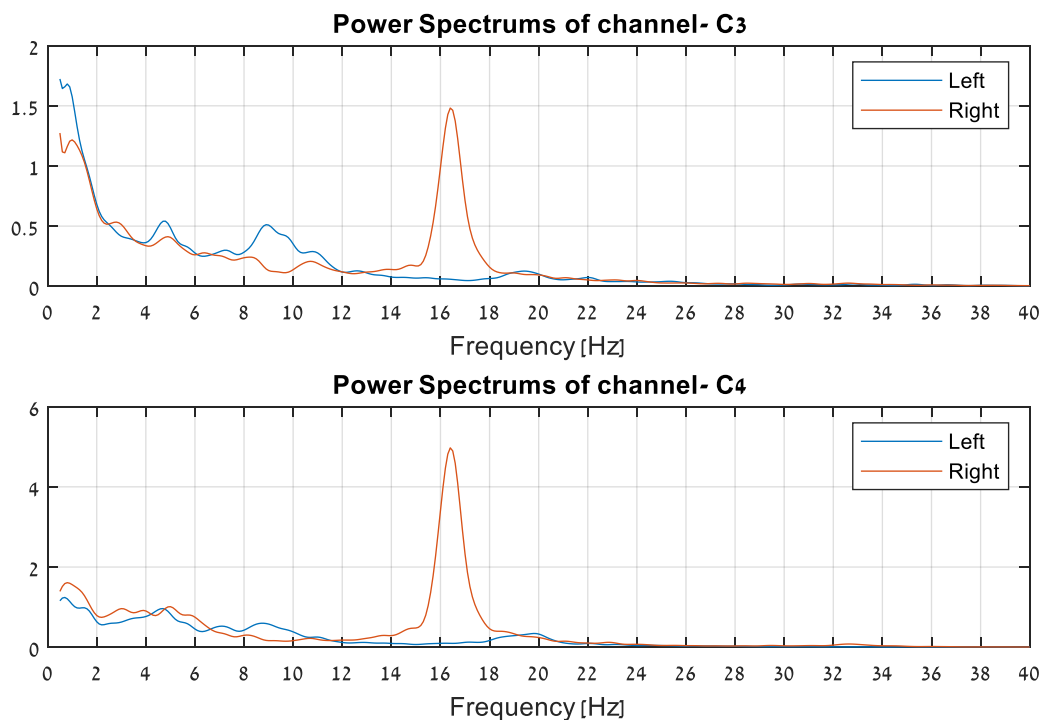


תוצאות הפונקציה plotChannelPwelch –

נפרט עבור כל ערוץ את הממצאים: [איור בעמוד הבא]

C3- עבור ערוץ זה באופן בולט העוצמה של מחלקה ימין חזקה משמאל בתחום 15-18 Hz. כמו כן בתחומים 3.5-6 Hz וגם 7-12 Hz העוצמה של מחלקה שמאל חזקה במעט משל מחלקה ימין.

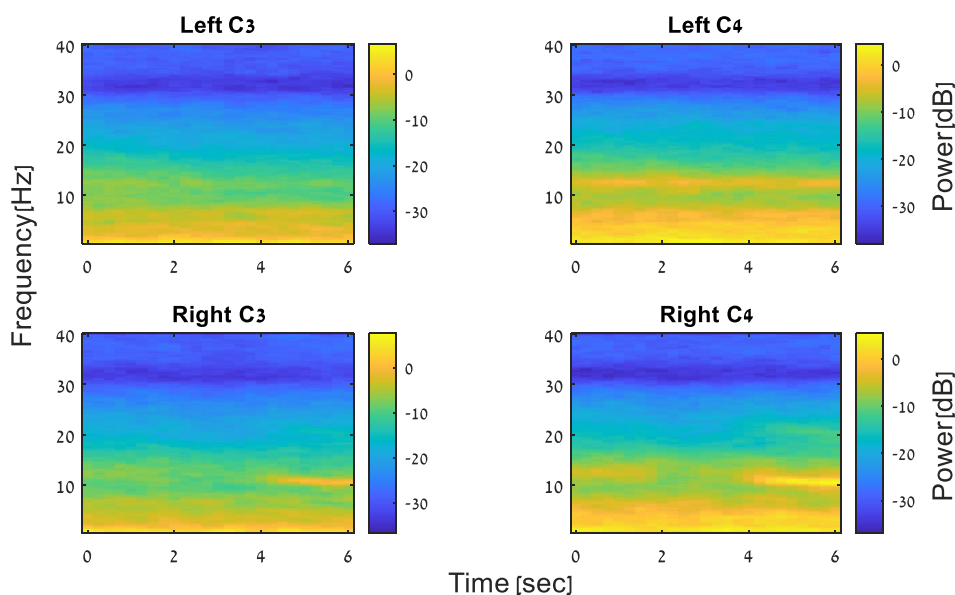
C4- עבור ערוץ זה באופן בולט העוצמה של מחלקה ימין חזקה משמאל בתחום 15-18 Hz. כמו כן בתחום 1-12 Hz ישנם הפרשים קטנים בין עוצמות המחלקות המשתנים לסירוגין.



תוצאות הפונקציה plotSpectrogram

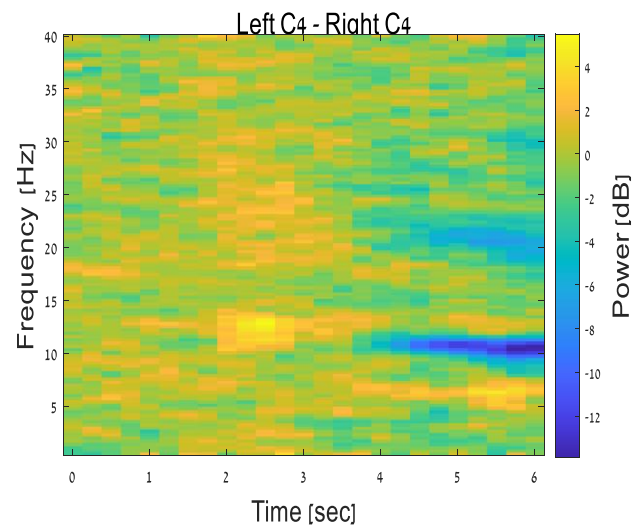
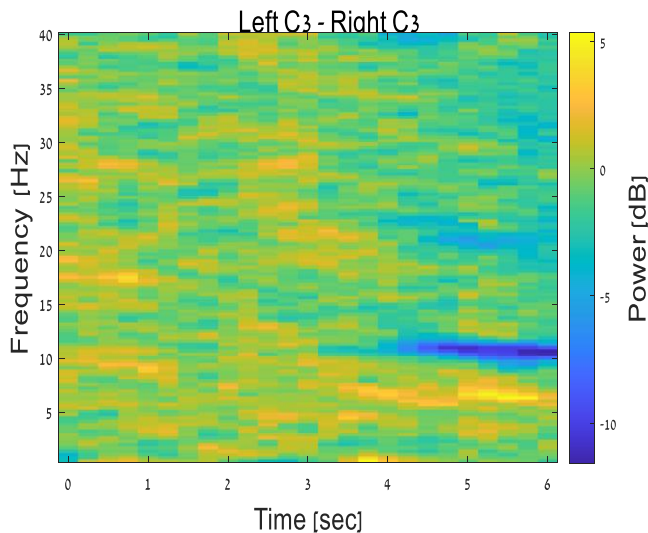
עבור כל המחלקות והערוצים, ישנה עוצמה חזקה בתחום עד 5 Hz ועוצמה חלשה החל מ-17 Hz. ניתן להבחין כי יש פעילות מוגברת בשני הערוצים במחלקה ימין בתחום 10-12 Hz החל משנייה 4. בנוסף בערוץ C4 במחלקה זו ישנה דעיכה קלה בעוצמת תחום 10-12 Hz בין 2-4 שניות. במחלקה שמאל ישנה פעילות חזקה יותר של ערוץ C4 בתחומים 10-12 Hz וגם 0-7 Hz לעומת ערוץ C3 לאורך כל משך הניסוי (6 שניות).

Spectrograms



תוצאות הפונקציה `plotSpecsDiff`

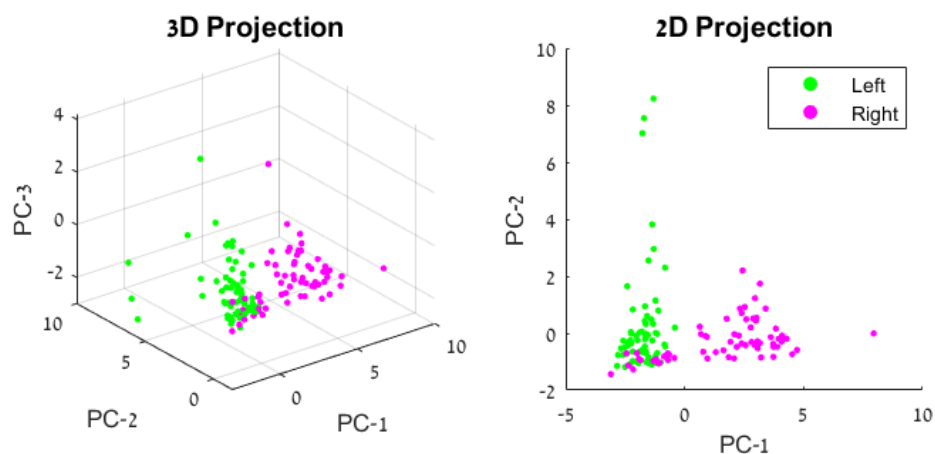
כדי להבין טוב יותר את הנתונים ולהוציא תכונות מתאימות נתבונן בהפרשי הספקטוגרמות עבור כל ערוץ. עבור ההפרש בין מחלקה שמאל לימין בערוץ C3 נזהה בנוסף עוצמה חזקה בתחום 6-8 Hz בזמן 5-6 שניות. עבור ההפרש בין מחלקה שמאל לימין בערוץ C4 נבחין בנוסף להפרש גבוה בתחומים 10-13 Hz ובתחום 20-27 Hz בטווח זמנים שבין 2-3 שניות ועוצמה גבוהה בתחום 5-7 Hz בשניות 5-6.



תוצאות הפונקציה `plotPca`

בהצגה הדו ממדית ניתן להבחין כי ישנה הפרדה ברורה בין המחלקות השונות, ישנו מקבץ עבור מחלקה ימין (0-5 בציר PC-1) בעוד שמחלקה שמאל מפוזרת בערכים נמוכים יותר בציר PC-1. יתר על כן בהצגה התלת ממדית ניתן ליצור חיץ נוסף בין המחלקות גם בעזרת הציר הנוסף PC-3.

Pca



תוצאות הפונקציה `plotFeaturesHistogram`

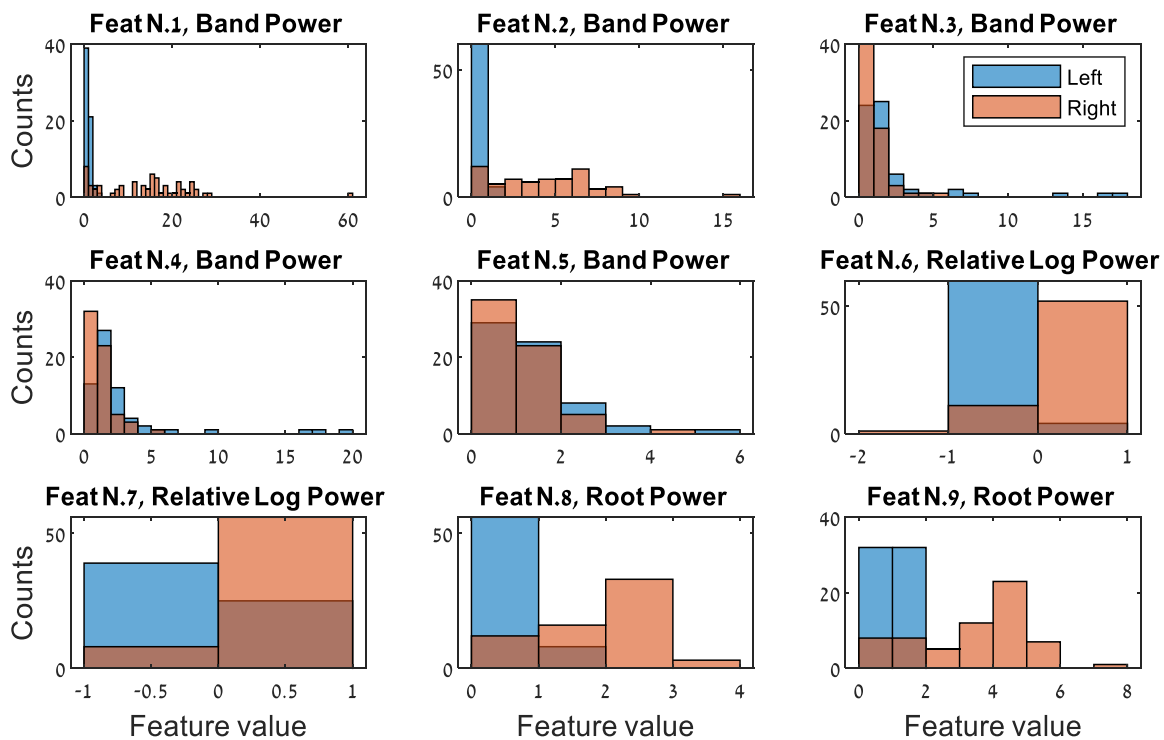
נרצה לבדוק האם תכונות מסוימת אכן הצליחה להבחין בין המחלקות השונות, נתבונן בממצאים לפי סוג התכונות:

Band power - עבור 2 התכונות הראשונות (בתחום 15-18 Hz) קיימת הפרדה ברורה - כל ה trials ממחלקה שמאל במקטעים 0-2 בעוד שמחלקה ימין מפוזרת על פני מקטעים שונים. 2 התכונות הנוספות מסוג זה לא הצליחו להפריד כראוי בין המחלקות.

Relative Log Power - עבור 2 התכונות קיימת הפרדה ברורה בין המחלקות, רוב ה trials נמצאים במקטעים בעלי ערכים שונים של התכונות.

Root Power - עבור 2 התכונות קיימת הפרדה ברורה בין המחלקות, רוב ה trials של מחלקה שמאל במקטע בעל ערך נמוך של התכונות בעוד שמחלקה ימין מפוזרת על פני מקטעים שונים.

Histograms



– Classification

נציג טבלת תוצאות אימון התכונות (features) על 128 ניסיונות (trials) האימון לאחר סיווג:

| תכונות (feature) | תכונות 9 | כל התכונות (1-9) |
|------------------|--------------------|--------------------|
| אימון training | $87.5 \pm 0.62\%$ | $89.14 \pm 0.96\%$ |
| בדיקה Validation | $87.37 \pm 6.93\%$ | $87.44 \pm 8.53\%$ |

ישנו דמיון רב בין תוצאות הרצת תכונות בודדת- תכונות 9 שחושבה על root power בערוץ C4 בתחום 15-18 Hz לבין תוצאות הרצת כל התכונות (1-9).
תכונות בודדת הצליחה לסווג ברמת דיוק מעט נמוכה מכל התכונות עם סטיות תקן נמוכות במעט מכל התכונות. כלומר בשני המקרים הסיווג למחלקות בוצע בדיוק בסדר גודל דומה.

תוצאות נתוני המבחן מהMoodlen :

כאשר הרצנו את הסיווג עבור נתוני המבחן (testData) עבור אותם פיצ'רים בדיוק (כל התשעה) קיבלנו **דיוק של 96.88**.

דיון ומסקנות:

השערנו הייתה שנוכל למצוא תכונות אשר יפרידו את נתוני הקלטת דמיון פעילות מוטורית למחלקות ימין ושמאל ואכן התוצאות איששו זאת. במהלך אימון המסווג קיבלנו אחוזי דיוק גבוהים (בסביבות 90 אחוזים) בעוד שהצלחת המבחן הייתה גבוהה אף יותר (96.88 אחוזים). ניתן להסיק כי אכן למדנו באופן מוצלח וגילינו תכונות אשר מסווגות נכונה את הנתונים. מניתוח הנתונים מיקדנו את התכונות בתחום ספציפי בעזרת תכונות מסוגים שונים, וגילינו כי תחום קריטי לסיווג תנועת ידיים (בעיקר עבור מחלקה ימין) הוא תחום התדרים 15-18 Hz.

השתמשנו בתהליך הסיווג בכל התכונות ולא רק בתכונות בודדת (למרות תוצאות דומות) כי שיערנו שאם נתייחס גם לתחומי תדרים שונים נלמד לזהות הבדלים בין המחלקות בצורה כללית עבור סט דוגמאות שונה משלנו.

יתכן כי מספר דגימות הבדיקה היה קטן ולכן הצלחנו ללמוד את דוגמאות המבחן במקריות בהצלחה גבוהה זו, לכן יש מקום למחקרי המשך אשר יחקרו את השערנו.

ניתן להסיק מהתוצאות, כי כלי הוויזואליזציה הוא מרכיב חשוב בלמידה והבנת motor imagery. לדעתנו **חשיבות יתרה** לניתוח power spectrum בדרכים השונות על הנתונים הגולמיים, כלי אשר עזר לנו לבודד תחומי תדר משמעותיים ומרכזיים מתוך ההקלטה אשר איששו את האבחנה בין המחלקות השונות.

לסיום ראינו כי ממשחק מחשב מוח הינו כלי יעיל ללמידה, על כן יש להמשיך ולפתח כלי זה אשר עשוי לעזור לנו בתחומים השונים- רפואה, ביטחון, תחבורה ותחומים רבים בחינוך.