

بسم تعالی



تمرین امتیازی پردازش سیگنال الکترومایوگرام به منظور شناسایی حرکات دست
اصول مهندسی پزشکی

امیرحسین زاهدی ۹۹۱۰۱۷۰۵

دکتر مهران جاهد

پاییز ۱۴۰۲

شرح تہرین:

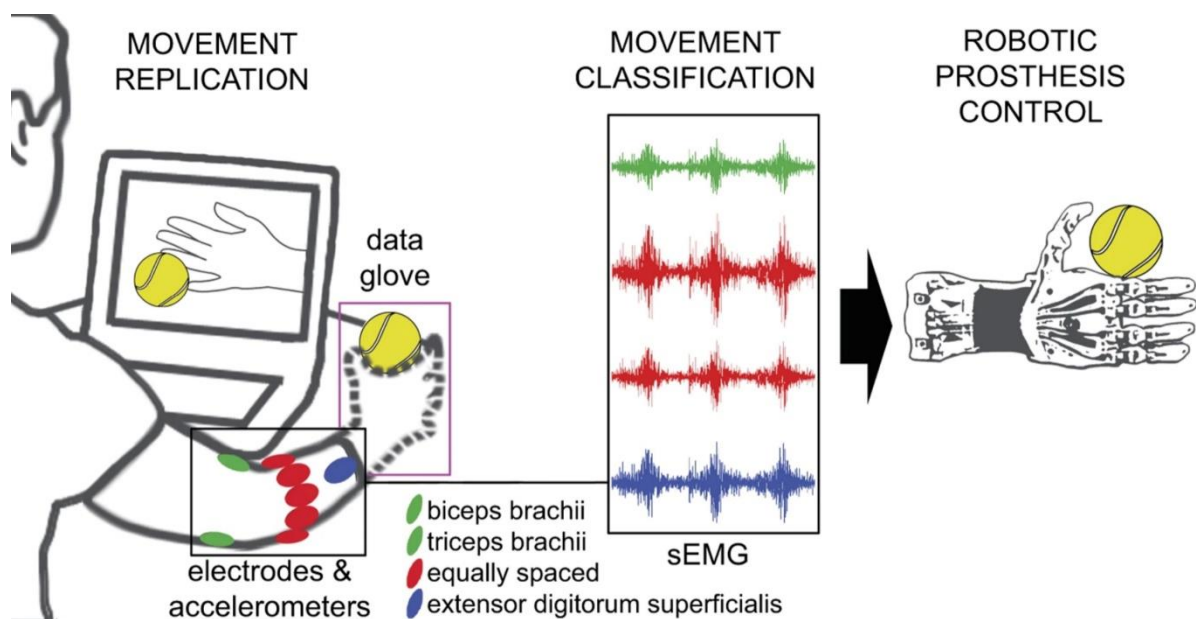
سیگنال EMG سیگنالی است که از فعالیت عضلات بدن بدست می آید و می تواند توسط حسگرها ضبط و پردازش شود. از این سیگنال می توان برای برآورد رفتار ماهیچه ها در حرکات بدن استفاده کرد. یکی از مجموعه رفتارهای مهم در بدن انسان، حرکت دست ها به خصوص میچ دست در هنگام گیرش اشیاء متنوع است. در این تمرین قصد داریم تا با استفاده از سیگنال های EMG ثبت شده، نوع حرکت انجام شده توسط دست شخص را متوجه شویم.

در این تمرین از داده های سه سابیجکت اول دیتابیس DB۲ در سایت Ninapro استفاده شده است زیرا که سیگنال ضبط شده توسط گروه در مرحله قبل همانطور که در پیوست نیز مشاهده می شود، مناسب انجام این تمرین نبوده است.

در این دیتا بیس دیتای سه دسته آزمایش از ۴۰ سابجکت موجود است که در این آزمایش ها هر کدام ۴۹ حرکت را به تعداد ۶ دفعه ۵ ثانیه ای انجام داده اند. در این دیتا بیس دیتای های متنوعی اعم از نیروسنج، شتابسنج و EMG وجود دارد.

[illegible]

در بخش EMG برای انجام هر حرکت از ۱۲ سنسور برای ثبت استفاده است که سنسورهای ۱ تا ۸ به صورت آرایه ای بر دور قسمت بالایی ساعد قرار گرفته اند، همچنین دو سنسور نیز در قسمت نزدیک مچ و در نهایت دو سنسور در قسمت باز قرار گرفته اند. با استفاده از این ۱۲ سنسور، EMG حاصل از ۴۹ حرکت در سه دسته با پیچیدگی های متنوع ضبط شده است. در این تمرین از دسته EX۱ سه سابجکت اول استفاده شده است که هر کدام شامل ۱۷ حرکت مچ دست هستند. حرکات خواسته شده در قسمت های قبل آزمایشگاه که شامل باز شدن دست، بسته شدن دست، بالا بردن مچ و پایین بردن مچ هستند به ترتیب حرکات شماره ۵، ۶، ۱۴ و ۱۳ هستند در این دسته ۱۷ تایی هستند. به جهت انجام با کیفیت تر این پروژه، در مسئله طبقه بندی هر ۱۷ حرکت منظور شده اند.



به صورت کلی در ابتدا دیتای موجود از ۱۲ کانال EMG را برای هر فرد پیش پردازش می کنیم، یعنی اینکه فیلتر های ناچ و بندپس بر روی آن اعمال می کنیم و آن ها را نرمالیزه می کنیم. سپس در مرحله بعد پنجره گذاری ها را بدست می آوریم به طوری که هر پنجره ۲۰۰ میلی ثانیه طول داشته باشد و ۱۹۰ میلی ثانیه با پنجره بعدی تداخل داشته باشد. پس از بدست آوردن پنجره ها ویژگی ها را استخراج می کنیم و در نهایت دیتا را به دو دسته ترین و تست تقسیم می کنیم. با اعمال طبقه بند بر روی دیتای تست با استفاده از لیبل های دیتای ترین شده، دیتای تست را طبقه بندی کرده و با واقعیت مقایسه می کنیم تا برآوردی از میزان دقت عملکرد داشته باشیم. موارد گفته شده در بالا به ترتیب در گزارش می آیند.

در انجام این تمرین از نرم افزار متلب استفاده شده است، البته برای مسائل یادگیری ماشین استفاده از پایتون بهتر است اما به دلیل راحتی بیشتر با نرم افزار متلب، متلب انتخاب شده است. الگوریتم و دیتای مورد استفاده قرار گرفته نیز در بالا توضیح داده شد.

بخش الف:

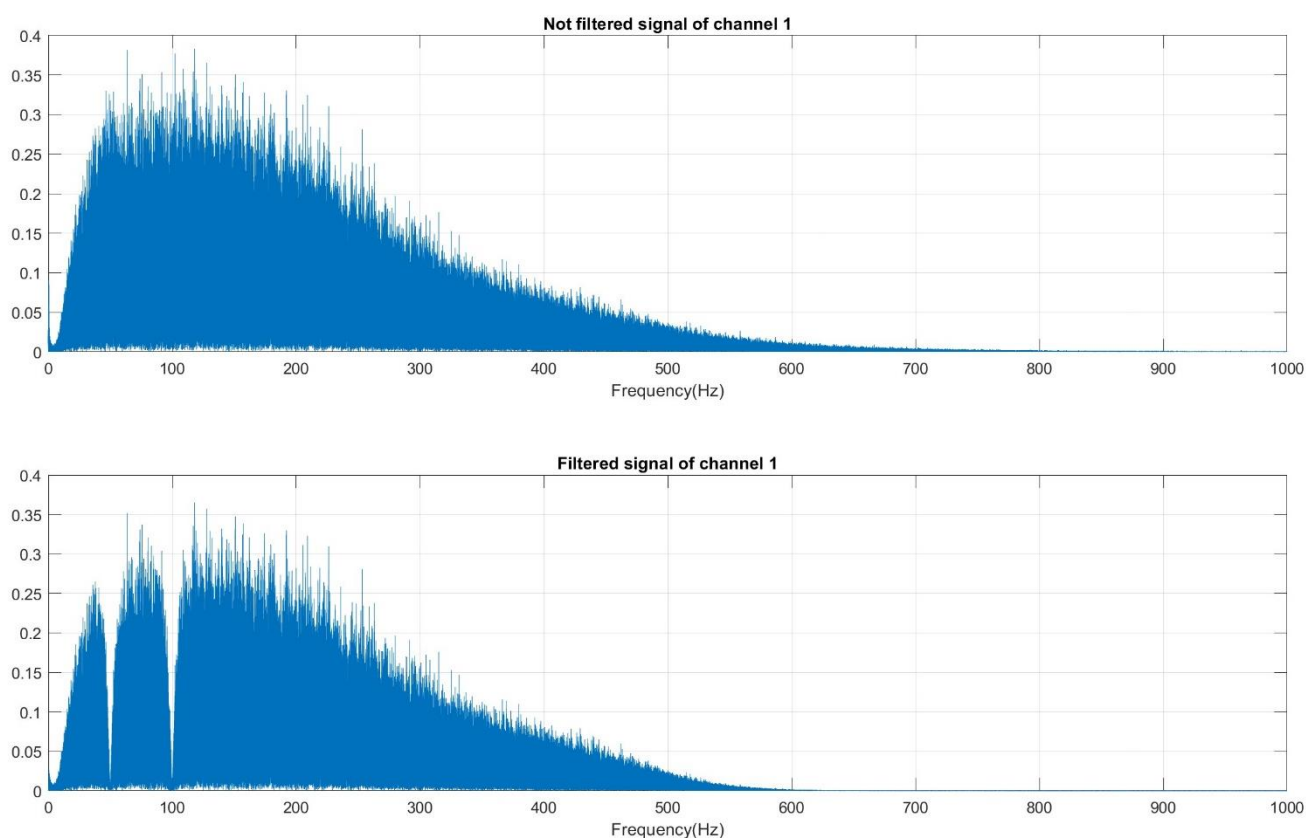
ابتدا دیتا ها را برای سابجکت مورد نظر لود می کنیم. سپس از دیتای EMG دوازده کاناله استفاده می کنیم و فیلترهای بند پس و ناچ را بر روی هر کانال به صورت جداگانه اعمال می کنیم.

فیلتر ناچ ۵۰ هرتز و هارمونیک اول آن ۱۰۰ هرتز را حذف می کند. و فیلتر بندپس، سیگنال های بالای ۱ هرتز و پایین ۵۰۰ هرتز را فیلتر می کند. به صورت کلی چون فرکانس نمونه برداری ۲ کیلو هرتز است، فرکانس مکس ۱ کیلو هرتز است و می توان با اطمینان خوبی گفت که طبق طیف فرکانسی مشاهده شده، فرکانس های بالای ۵۰۰ هرتز محتوای مفید ندارند و بهتر است تضعیف یا حذف شوند.

از فیلتر باتروورث مرتبه ۶ و ناچ استفاده می کنیم:

```
[b1,a1] = butter(6,500/(fs/2),'low'); % Butterworth lowpass filter for 500Hz  
[b2,a2] = butter(2,1/(fs/2),'high'); % Butterworth highpass filter for 1Hz  
[b3,a3] = iirnotch(50/(fs/2), 0.01); % Notch filter for 50Hz  
[b4,a4] = iirnotch(100/(fs/2), 0.01); % Notch filter for 100Hz
```

طیف فرکانسی سیگنال های قبل و بعد از فیلتر از کانال اول به شکل زیر هستند.



مشاهده می کنیم که فیلتر بند پس توانسته فرکانس های زیر ۱ هرتز و به خصوص فرکانس های بالاتر از ۵۰۰ هرتز را تا حد بسیار خوبی فیلتر کند همچنین فیلتر های ناچ ۵۰ و ۱۰۰ هرتز نیز به وضوح نویز برق شهر را حذف کرده اند.

بخش ب:

در این مرحله داده ها را به وسیله قطعه کد زیر در هر کانال نرمالایز می کنیم به صورتی که میانگینشان ۰ و واریانس برابر ۱ شود. (Z-Score)

```
for i = 1:12
    normalized_emg(:,i) = normalize(filtered_emg(:,i), 'zscore');
end
```

همه کانال ها را نرمالیزه می کنیم اما دیتا های تست و ترین را در این مرحله جدا نمی کنیم و پس از پنجره بندی این کار را انجام می دهیم زیرا که پس از پنجره بندی تعداد زیادی پنجره با لیبیل های مشخص بدست می آید که میتوانیم با تعیین کردن رندوم پنجره های ترین و پنجره های تست، طبقه بندی را انجام دهیم.

بخش ج:

در این بخش دیتای بزرگ EMG که دوازده کانال دارد و هر کانال در حدود ۲ میلیون سمپل است را پنجره بندی می کنیم. همانطور که در بخش اول نیز گفتیم این سیگنال حاصل از انجام ۱۷ حرکت دست هر کدام با تکرار ۶ بار و هر تکرار به طول ۵ ثانیه انجام و ۳ ثانیه استراحت است. در این داده ۱۷ نوع حرکت به ترتیب از ۱ تا ۱۷ هر کدام شش بار تکرار شده اند. لیبیل هر سمپل نیز مشخص است.

برای پنجره بندی این سیگنال، پنجره های ۴۰۰ نمونه ای معادل با ۲۰۰ میلی ثانیه در نظر می گیریم که با یکدیگر ۱۹۰ میلی ثانیه یا به عبارتی ۳۸۰ سمپل همپوشانی دارند.

برای مثال برای سابجکت اول حدود ۹۰ هزار پنجره دیتا بدست می آید. لیبیل مخصوص هر پنجره را با پیدا کردن بیشترین لیبیل های موجود در دیتاهای هر پنجره مشخص می کنیم. مثلاً اگر لیبیل ۱۲ در پنجره ۵۶۳۸۱ بیشترین تعداد را داشته باشد، لیبیل آن پنجره ۱۲ است. به دلیل تعداد بسیار زیاد پنجره های با لیبیل های ۰ که نشان دهنده استراحت هستند و نا متعادل کردن دیتا، پنجره های با لیبیل ۰ را حذف می کنیمو اینگونه در حدود ۵۰ هزار پنجره لیبیل خورده باقی می ماند.

خروجی این بخش ماتریس $12 \times 400 \times N$ است که ۴۰۰ نمونه های زمانی و ۱۲ تعداد کانال و N تعداد پنجره های غیر لیبیل ۰ است. البته وکتور لیبیل های هر پنجره نیز به عنوان خروجی بدست می آید.

بخش د:

در این بخش ۷ ویژگی مقدار مطلق متوسط، انحراف معیار، واریانس، طول موج سیگنال، گذر از صفر، مقدار ریشه میانگین و مقدار مطلق یکپارچه را به عنوان ویژگی های هر کانال بر می گزینیم و حساب می کنیم.

در نهایت نیز چون ۱۲ کانال داریم و از هر کانال ۷ ویژگی بدست آورده ایم، ماتریس ویژگی ها را تولید می کنیم که به تعداد پنجره ها سطر دارد و به تعداد ویژگی ها ستون دارد که تعداد این ویژگی ها برابر با $12 \times 7 = 84$ ویژگی است. در این بخش خروجی این ماتریس مهم ویژگی ها است.

بخش ۵:

با توجه به ماتریس های ویژگی ها و لیبل ها، در ابتدا با استفاده از تابع `cvpartition` ۷۰ درصد پنجره ها را به عنوان دیتای آموزش ذخیره می کنیم و ۳۰ درصد از دیتا را به عنوان دیتای تست در نظر می گیریم که در کد نیز به خوبی تفکیک شده اند.

مدل های `knn` و `random forest` را انتخاب می کنیم. برای `knn` از تابع زیر با در نظر گرفتن نزدیکی ۵ همسایه استفاده می کنیم.

```
fitcknn(data_train,label_train,'NumNeighbors',5,'Distance','euclidean');
```

برای مدل `rf` نیز از تابع زیر با در نظر گرفتن ۱۰۰ درخت تصمیم استفاده می شود.

```
fitcknn(data_train,label_train,'NumNeighbors',5,'Distance','euclidean');
```

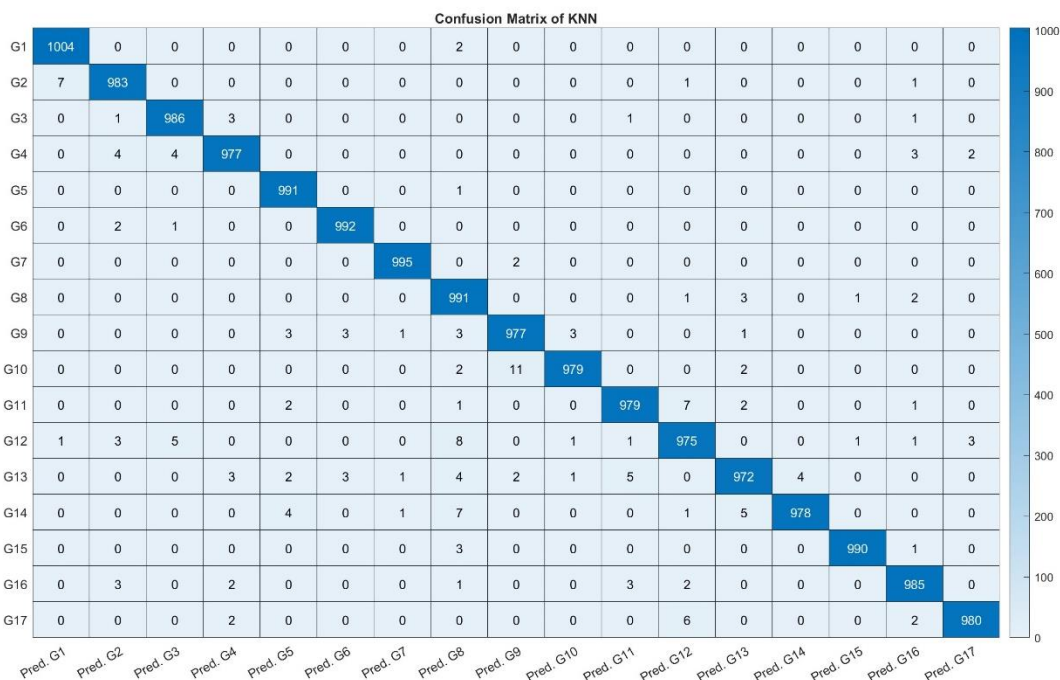
در نهایت با استفاده از مدل های ترین شده از هر دو روش و به وسیله دادن مدل ها و پنجره های تست به تابع `predict`، لیبل داده های تست را تعیین می کنیم.

پس از تعیین لیبل داده های تست برای صحت سنجی طبقه بندی انجام شده به سراغ ماتریس کانفیوژن و `accuracy` بدست آمده از آن می رویم.

در این بخش دیتا را برای هر کدام از سابجکت ها به صورت جدا گانه طبقه بندی کرده و خروجی ها را بدست می آوریم.

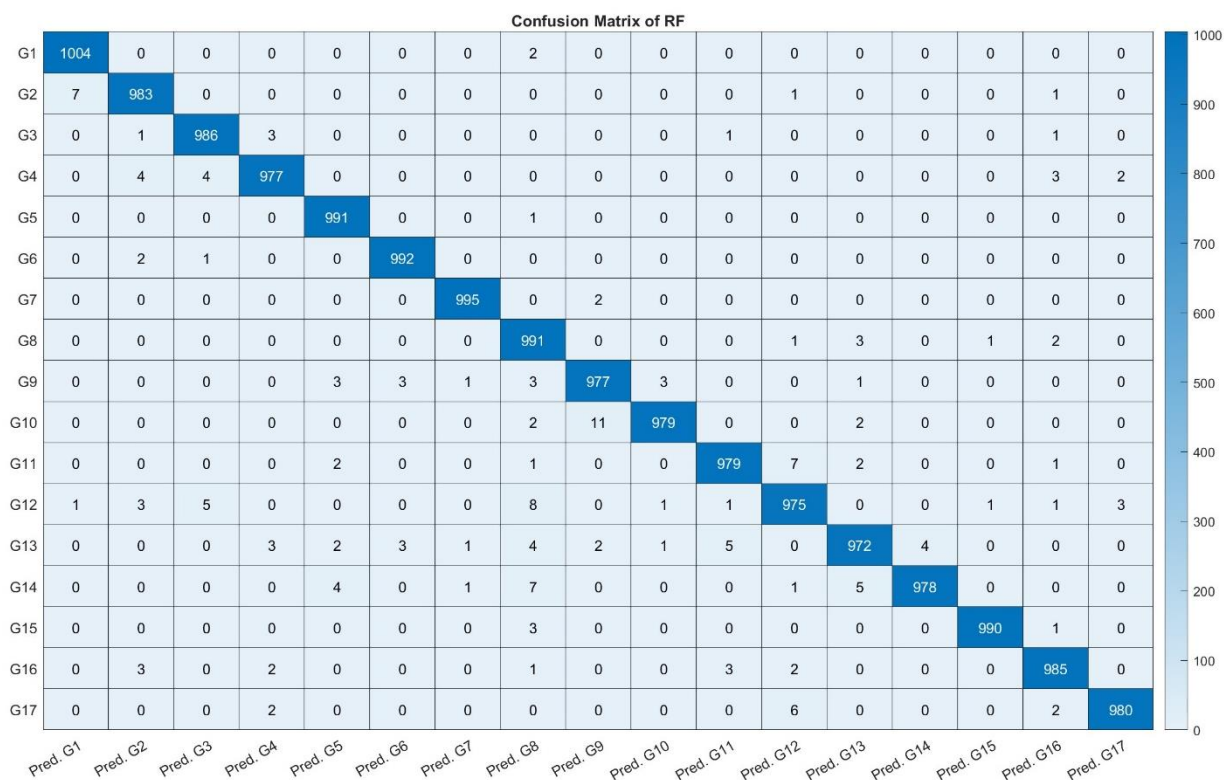
سابجکت اول:

مدل `knn`:



"Accuracy with KNN model is : ۹۸.۹۵۳۳%"

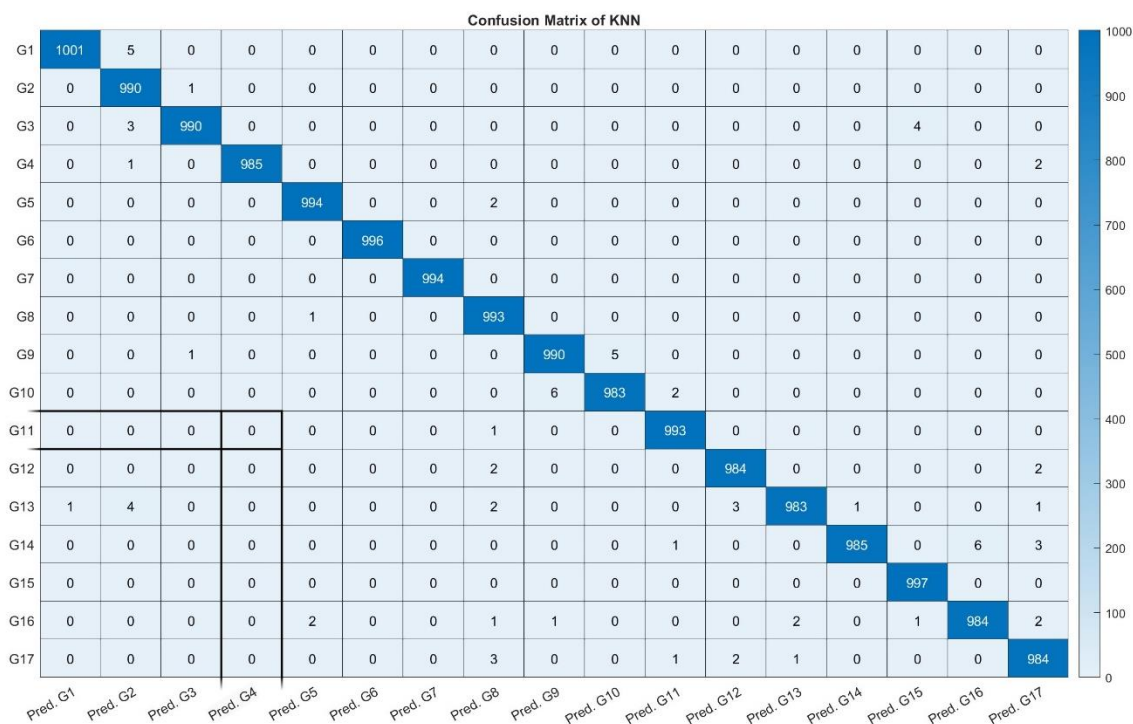
مدل rf:



"Accuracy with RF model is : ۹۹.۳۹۰۹%"

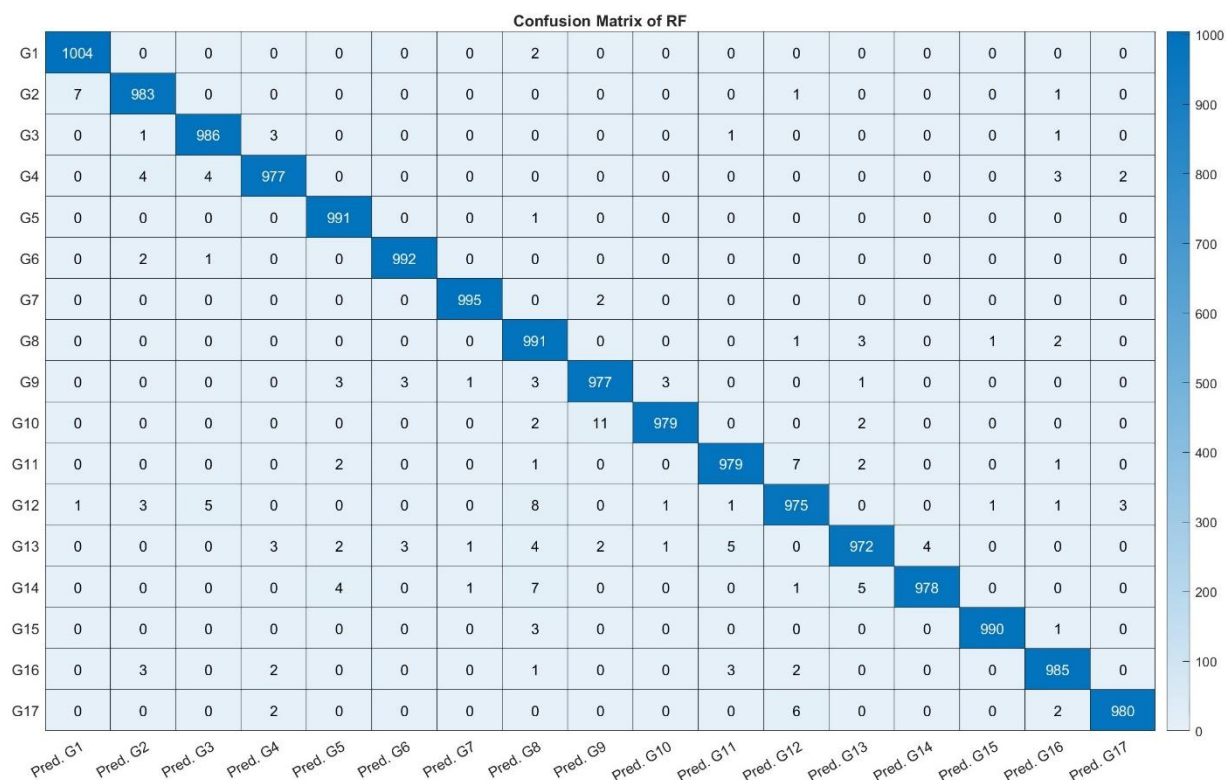
سابجکت دوم:

مدل knn:



"Accuracy with KNN model is : ۹۹.۵۵۰۳%"

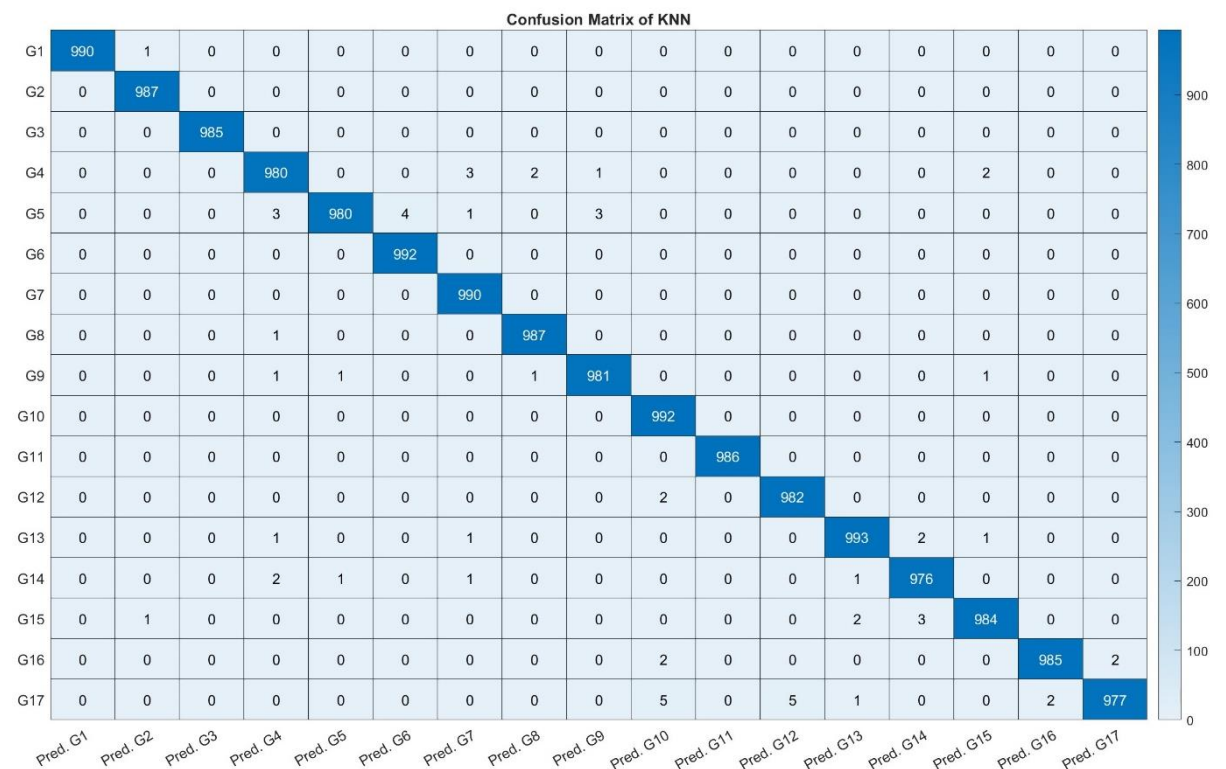
مدل rf:



"Accuracy with RF model is : ۹۹.۵۳۸۵%"

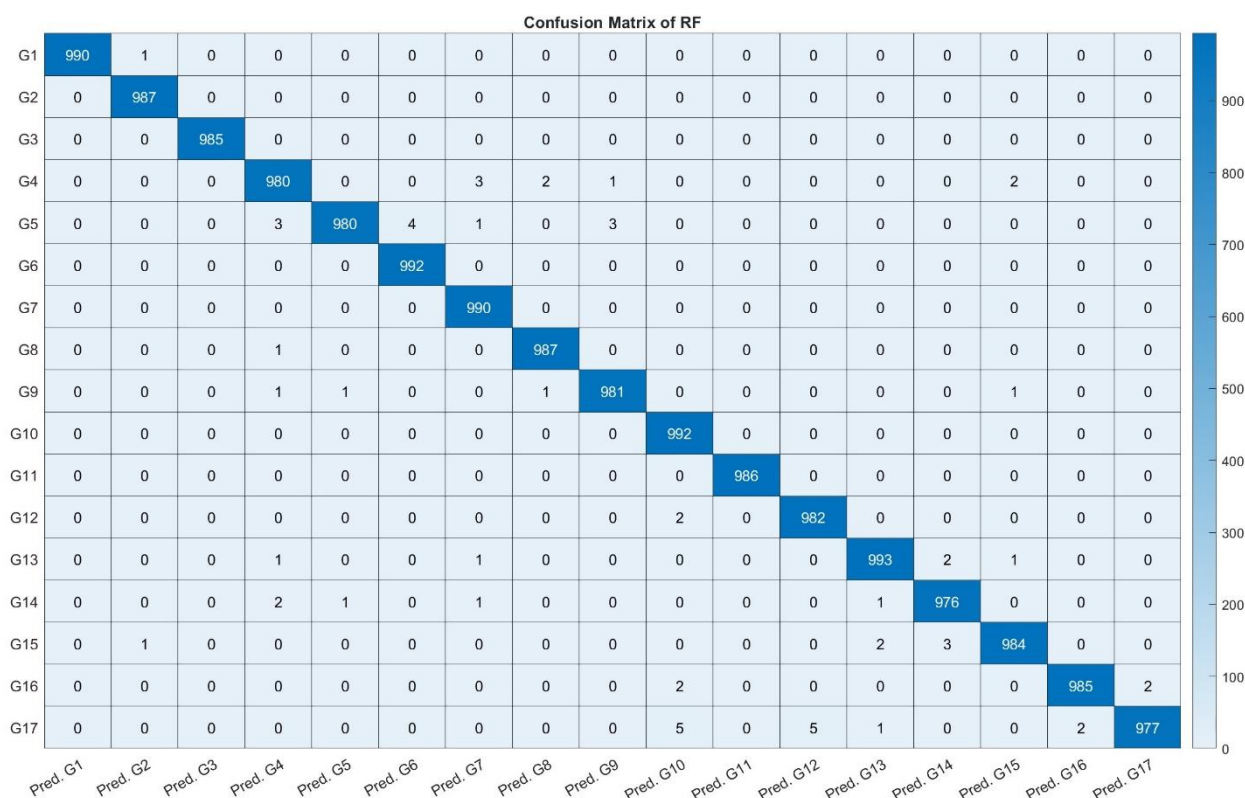
سابجکت سوم:

مدل knn:



"Accuracy with KNN model is : ۹۹.۶۴۳%"

مدل rf:



"Accuracy with RF model is : 99.5181%"

همانطور که از ماتریس های کانفیوژن و دقت های بدست آمده در هر دو مدل برای هر سه سابجکت دیده می شود، طبقه بندی بطرز بسیار خوب و حیرت انگیزی انجام شده است. این نتیجه گیری بسیار خوب احتمالا به دلیل حضور ۱۲ کانال با همان سنسور در نقاط مختلف دست و همچنین در نظر گرفتن تعداد مناسبی ویژگی است.

بخش ی:

از جمله چالش های این تمرین استفاده از دیتابیس آماده و به نسبت بزرگ بود که اتفاقاً بر جذابیت کار بسیار می افزود. اینکه به جای صرفاً ۴ حرکت تمام ۱۷ حرکت را بخواهیم با ۷ ویژگی جدا کنیم چالش بزرگی بود که به خوبی انجام شد.

می‌توان برای بهبود تمرین حرکت‌های پیچیده تر و دیتا‌های ساجکت‌های بیشتر را در نظر گرفت تا قدرت طبقه‌بندی مطرح شده را بتوان بهتر مورد سنجش قرار داد.

در کل تمرین آموزنده و زیبایی بود، بسیار ممنون.

منبع دیتاست:

[Ninapro \(hevs.ch\)](http://hevs.ch)