

```
function [count1,count2,n1] = ac_intention1(movement,neutral);

% this gets all trials for a single channel and classifies between neutral
% and movement and returns the classification rate or percentage. For this
% it takes 500ms of data when there is no movement and it takes 500ms
% before the movement is initiated.
%%%%%%%%%%%%%%%
%%
%%%%%
%%%baseline correction
n = size(movement,2);

for i = 1:n
    neutral(:,i) = neutral(:,i) - mean(neutral(:,i));
    movement(:,i) = movement(:,i) - mean(movement(:,i));
end

clear t1 t2 mid

%%%%%%%%%%%%%%%
%%
%split into training and testing data sets and compute PCA(U) and average
%feature vectors for each dataset.
t1 = movement;
t2 = neutral;

t1 = t1';
t2 = t2';

tr1 = t1(2:2:n,:);
tr2 = t2(2:2:n,:);
te1 = t1(1:2:n,:);
te2 = t2(1:2:n,:);

t = [tr1;tr2];

[U, lambda, tr] = pcaint1(t');
%size(U);
%U = U(:,1:10);
%lambda = lambda(1:10);

ptr1 = tr1*U;
ptr2 = tr2*U;
pte1 = te1*U;
pte2 = te2*U;

m1 = mean(ptr1,1);
m2 = mean(ptr2,1);

n1 = size(pte1,1);

%%%%%%%%%%%%%%%
%%
%classification.
```

```

vpte1 = 1;
vpte2 = 2;

pos = [1 2];
count1 = 0;
for i = 1:n1
diff1 = (pte1(i,:)-m1).* (pte1(i,:)-m1).*lambda';
diff2 = (pte1(i,:)-m2).* (pte1(i,:)-m2).*lambda';
res(1) = sum(diff1);
res(2) = sum(diff2);
if pos(find(res == min(res))) == vpte1
count1 = count1 + 1;
end
end
count1;
clear diff1 diff2 res count pos

pos = [1 2];
count2 = 0;
for i = 1:n1
diff1 = (pte2(i,:)-m1).* (pte2(i,:)-m1).*lambda';
diff2 = (pte2(i,:)-m2).* (pte2(i,:)-m2).*lambda';
res(1) = sum(diff1);
res(2) = sum(diff2);
if pos(find(res == min(res))) == vpte2
count2 = count2 + 1;
end
end
count2;
clear diff1 diff2 res count pos

```