```
L = load('MNIST_20x20.mat');
labels = L.labels;
imgs = L.imgs;
n = size(labels,1);
counter = zeros(4);
for i = 1:n
    if labels(i) == 1
        counter(1) = counter(1) + 1;
        IMG(counter(1),1) = i;
    elseif labels(i) == 2
        counter(2) = counter(2) + 1;
        IMG(counter(2),2) = i;
    elseif labels(i) == 3
        counter(3) = counter(3) + 1;
        IMG(counter(3),3) = i;
    elseif labels(i) == 4
        counter(4) = counter(4) + 1;
        IMG(counter(4),4) = i;
    end
end

% pair: 3 & 4
k = 3;
l = 4;
total = counter(k)+counter(l);
X = zeros(400,total);
Y = (total);
for i = 1:counter(k)
    X(1:400,i) = reshape(imgs(:,:,IMG(i,k)),[400,1]);
    Y(i) = 1;
end
for i = 1:counter(l)
    j = counter(k)+i;
    X(1:400,j) = reshape(imgs(:,:,IMG(i,l)),[400,1]);
    Y(j) = -1;
end
X = transpose(X);

PC = pca(X);
mean = ones(1,total)*X/total;
proj = zeros(total,2);
for i = 1:total
    proj(i,1) = (X(i,:)-mean)*PC(:,1);
    proj(i,2) = (X(i,:)-mean)*PC(:,2);
end

% % Hard SVM
% SVM_hard = fitcsvm(X,Y,'BoxConstraint',Inf);
% sv = SVM_hard.SupportVectors;
% figure;
% gscatter(proj(:,1),proj(:,2),Y);
% hold on;
% % plot(sv(:,1),sv(:,2),'ko','MarkerSize',10);
% % legend('-1','1','Support Vector');
% new_beta = SVM_hard.Beta'*PC(:,1:2);
% m = - new_beta(1)/new_beta(2);
% b = - SVM_hard.Bias/new_beta(2);
% refline(m,b);
% margin = 1/new_beta(2);
% l1 = refline(m,b+margin);
% l2 = refline(m,b-margin);
% l1.LineStyle = '--';
% l2.LineStyle = '--';
% hold off;
% count = 0;
% for i = 1:total
%     crit = X(i,:)*SVM_hard.Beta(:,1)+ SVM_hard.Bias;
%     if Y(i)*crit > 0
%         count = count + 1;
%     end
% end
% accuracy = count/total;

% % Soft SVM
% for i = 1:4
%     BC = 0.04+(i-1)*0.005;
%     SVM_soft = fitcsvm(X,Y,'BoxConstraint',BC);
%     CVSVM_soft = crossval(SVM_soft,'KFold',2);
%     Loss_soft = kfoldLoss(CVSVM_soft,'LossFun','classiferror') % average loss
% end
% Best BC = 0.04
% SVM_soft = fitcsvm(X,Y,'BoxConstraint',0.04);
% sv = SVM_soft.SupportVectors;
% figure;
% gscatter(proj(:,1),proj(:,2),Y);
% hold on;
% % plot(sv(:,1),sv(:,2),'ko','MarkerSize',10);
% % legend('-1','1','Support Vector');
% new_beta = SVM_soft.Beta'*PC(:,1:2);
% m = - new_beta(1)/new_beta(2);
% b = - SVM_soft.Bias/new_beta(2);
% refline(m,b);
% margin = 1/new_beta(2);
% l1 = refline(m,b+margin);
% l2 = refline(m,b-margin);
% l1.LineStyle = '--';
% l2.LineStyle = '--';
% hold off;
% count = 0;
% for i = 1:total
%     crit = X(i,:)*SVM_soft.Beta(:,1)+ SVM_soft.Bias;
%     if Y(i)*crit > 0
%         count = count + 1;
%     end
% end
% accuracy = count/total;

% SVM with Guassian kernel
SVM_Gaussian = fitcsvm(proj,Y,'BoxConstraint',↙
Inf,'KernelFunction','rbf','KernelScale','auto');
sv = SVM_Gaussian.SupportVectors;
figure;
gscatter(proj(:,1),proj(:,2),Y);
hold off;
% plot(sv(:,1),sv(:,2),'ko','MarkerSize',10);
% legend('-1','1','Support Vector');
% d = 2;
x1Grid = linspace(-6,7,27);
x2Grid = linspace(-6,7,27);
% [x1Grid,x2Grid] = meshgrid(min(proj(:,1)):d:max(proj(:,1)),min(proj(:,2)):d:max(proj(:,↙
2)));
% XGrid = [x1Grid(:),x2Grid(:)];
% [~,score] = predict(SVM_Gaussian,XGrid);
% % contour(x1Grid,x2Grid,reshape(score(:,2),size(x1Grid)),[0 0],'k');
% figure;
% gscatter(x1Grid,x2Grid,reshape(score(:,2),size(x1Grid)));
count = 0;
xx = zeros(size(x1Grid,2)*size(x2Grid,2),2);
p = zeros(size(x1Grid,2)*size(x2Grid,2),1);
for i = 1:size(x1Grid,2)
    for j = 1:size(x2Grid,2)
        count = count + 1;
        xx(count,:) = [x1Grid(i),x2Grid(j)];
        p(count,1) = predict(SVM_Gaussian,xx(count,:));
    end
end
figure;
gscatter(xx(:,1),xx(:,2),p(:,1));

crit = zeros(total);
count = 0;
for i = 1:total
    [~,crit(i)] = predict(SVM_Gaussian,proj(i,:));
    if Y(i)*crit(i) > 0
        count = count + 1;
    end
end
accuracy = count/total;
```