Part g)

this chunk of code prepares F_train, F_test y_train and y_test for all categories and save them in separate folders with class names

```
CD=cd;
categories=dir('101_ObjectCategories');
categories(1:2)=[];
for k=1:numel(categories)
    Directory=categories(k).name;
    Names1=dir(['101_ObjectCategories\' Directory '\']);
    Names1(1:2)=[];
    Names2=dir(['Annotations\' Directory '\']);
    Names2(1:2)=[];
    F_train=nan(floor(numel(Names1)*.9),106);
    y_train=cell(size(F_train,1),1);
    F_test=nan(numel(Names1)-size(F_train,1),106);
    y_test=cell(size(F_test,1),1);
    for i=1:size(F_train,1)
        im=double(imread([cd '\101_ObjectCategories\' Directory '\' Names1(i).name]))/255;
        [M,N,~]=size(im);
        ann=load([cd '\Annotations\' Directory '\' Names2(i).name]);
        mask=double(poly2mask(ann.obj_contour(1,:)+ann.box_coord(3),ann.obj_contour(2,:)+ann.box_coord(1),M,N));
        F_train(i,1:30)=extract_color_features(im*255,mask);
        F_train(i,31:40)=extract_boundary_features(mask);
        F_train(i,41:47)=extract_hu_moments(mask);
        F_train(i,48:58)=extract_props(mask);
        F_train(i,59:end)=extract_texture_features(im,mask);
        y_train{i}=Directory(1:end-1);
    end
    for j=i+1:numel(Names1)
        im=double(imread([cd '\101_ObjectCategories\' Directory '\' Names1(j).name]))/255;
        [M,N,~]=size(im);
        ann=load([cd '\Annotations\' Directory '\' Names2(j).name]);
        mask=double(poly2mask(ann.obj_contour(1,:)+ann.box_coord(3),ann.obj_contour(2,:)+ann.box_coord(1),M,N));
        F_test(j-i,1:30)=extract_color_features(im*255,mask);
        F_test(j-i,31:40)=extract_boundary_features(mask);
        F_test(j-i,41:47)=extract_hu_moments(mask);
        F_test(j-i,48:58)=extract_props(mask);
        F_test(j-i,59:end)=extract_texture_features(im,mask);
        y_test{j-i}=Directory(1:end-1);
    end
    [Fn_train,mx,mn]=normalize_feature_columns(F_train);
    Fn_test=normalize_feature_columns(F_test,mx,mn);
    mkdir([CD '\Features\' Directory])
    cd([CD '\Features\' Directory])
    save(['features_' Directory],'Fn_train','Fn_test','y_train','y_test')
    cd(CD)
end
```

i)

```
CD=cd;
Cat=["emu" "flamingo"];
F1=[];
y1=[];
F2=[];
y2=[];
for i=1:numel(Cat)
    cd([cd '\Features\' char(Cat(i)) '\'])
    load(['features_' char(Cat(i))])
    F1=[F1;Fn_train];
    F2=[F2;Fn_test];
    y1=[y1;y_train];
    y2=[y2;y_test];
    cd(CD)
end
Fn_train=F1;
Fn_test=F2;
y_train=y1;
y_test=y2;
clf=fitcsvm(Fn_train,y_train);
y_test_hat=clf.predict(Fn_test);
C=confusionmat(y_test,y_test_hat);
acc=sum(diag(C))/sum(sum(C));
disp('Confusion Matrix:')
```

```
Confusion Matrix:
```

```
disp(C)
```

```
     6     0
     0     7
```

```
disp('Accuracy (in percent):')
```

```
Accuracy (in percent):
```

```
disp(acc*100)
```

```
   100
```

ii)

```
CD=cd;
Cat=["emu" "flamingo" "strawberry"];
F1=[];
y1=[];
F2=[];
y2=[];
for i=1:numel(Cat)
    cd([cd '\Features\' char(Cat(i)) '\'])
    load(['features_' char(Cat(i))])
    F1=[F1;Fn_train];
    F2=[F2;Fn_test];
    y1=[y1;y_train];
    y2=[y2;y_test];
    cd(CD)
end
Fn_train=F1;
Fn_test=F2;
y_train=y1;
y_test=y2;
clf=fitcecoc(Fn_train,y_train);
y_test_hat=clf.predict(Fn_test);
C=confusionmat(y_test,y_test_hat);
acc=sum(diag(C))/sum(sum(C));
disp('Confusion Matrix:')
```

```
Confusion Matrix:
```

```
disp(C)
```

```
     6     0     0
     0     7     0
     0     0     4
```

```
disp('Accuracy (in percent):')
```

```
Accuracy (in percent):
```

```
disp(acc*100)
```

```
   100
```

iii)

```
CD=cd;
Cat=["emu" "flamingo"];
F1=[];
y1=[];
F2=[];
y2=[];
for i=1:numel(Cat)
    cd([cd '\Features\' char(Cat(i)) '\'])
    load(['features_' char(Cat(i))])
    F1=[F1;Fn_train];
    F2=[F2;Fn_test];
    y1=[y1;y_train];
    y2=[y2;y_test];
    cd(CD)
end
Fn_train=F1(:,1:30);
Fn_test=F2(:,1:30);
y_train=y1;
y_test=y2;
clf=fitcecoc(Fn_train,y_train);
y_test_hat=clf.predict(Fn_test);
C=confusionmat(y_test,y_test_hat);
acc=sum(diag(C))/sum(sum(C));
disp('Confusion Matrix using only color features:')
```

```
Confusion Matrix using only color features:
```

```
disp(C)
```

```
     6     0
     0     7
```

```
disp('Accuracy using only color features (in percent):')
```

```
Accuracy using only color features (in percent):
```

```
disp(acc*100)
```

```
   100
```

```

Fn_train=F1(:,31:40);
Fn_test=F2(:,31:40);
clf=fitcecoc(Fn_train,y_train);
y_test_hat=clf.predict(Fn_test);
C=confusionmat(y_test,y_test_hat);
acc=sum(diag(C))/sum(sum(C));
disp('Confusion Matrix using only boundary features:')
```

```
Confusion Matrix using only boundary features:
```

```
disp(C)
```

```
     0     6
     1     6
```

```
disp('Accuracy using only boundary features (in percent):')
```

```
Accuracy using only boundary features (in percent):
```

```
disp(acc*100)
```

```
   46.154
```

```

Fn_train=F1(:,41:58);
Fn_test=F2(:,41:58);
clf=fitcecoc(Fn_train,y_train);
y_test_hat=clf.predict(Fn_test);
C=confusionmat(y_test,y_test_hat);
acc=sum(diag(C))/sum(sum(C));
disp('Confusion Matrix using only region features:')
```

```
Confusion Matrix using only region features:
```

```
disp(C)
```

```
     5     1
     0     7
```

```
disp('Accuracy using only region features (in percent):')
```

```
Accuracy using only region features (in percent):
```

```
disp(acc*100)
```

```
   92.308
```

```

Fn_train=F1(:,59:end);
Fn_test=F2(:,59:end);
clf=fitcecoc(Fn_train,y_train);
y_test_hat=clf.predict(Fn_test);
C=confusionmat(y_test,y_test_hat);
acc=sum(diag(C))/sum(sum(C));
disp('Confusion Matrix using only texture features:')
```

```
Confusion Matrix using only texture features:
```

```
disp(C)
```

```
     6     0
     0     7
```

```
disp('Accuracy using only texture features (in percent):')
```

```
Accuracy using only texture features (in percent):
```

```
disp(acc*100)
```

```
   100
```

In this case using only color features and only texture features give 100 percent accuracy which makes sense, as two different birds differ obviously in color and texture, making color and texture most important features. region feature gives 92 percent accuracy, as a matter of fact one of the emus is cllasified as a flamingo. boundary features gives the worst accuracy of about 46 percent (all emus are classified as flamingos and one flamingo as an emu! hence using only region features causes the classifier to err tward flamingo!

iv)

```
CD=cd;
Cat=["chair" "windsor_chair"];
```

```
F1=[];
y1=[];
F2=[];
y2=[];
for i=1:numel(Cat)
    cd([cd '\Features\' char(Cat(i)) '\'])
    load(['features_' char(Cat(i))])
    F1=[F1;Fn_train];
    F2=[F2;Fn_test];
    y1=[y1;y_train];
    y2=[y2;y_test];
    cd(CD)
end
Fn_train=F1;
Fn_test=F2;
y_train=y1;
y_test=y2;
clf=fitcecoc(Fn_train,y_train);
y_test_hat=clf.predict(Fn_test);
C=confusionmat(y_test,y_test_hat);
acc=sum(diag(C))/sum(sum(C));
disp('Confusion Matrix using all features:')
```

```
Confusion Matrix using all features:
```

```
disp(C)
```

```
     7     0
     0     6
```

```
disp('Accuracy using all features (in percent):')
```

```
Accuracy using all features (in percent):
```

```
disp(acc*100)
```

```
   100
```

```

Fn_train=F1(:,1:30);
Fn_test=F2(:,1:30);
y_train=y1;
y_test=y2;
clf=fitcecoc(Fn_train,y_train);
y_test_hat=clf.predict(Fn_test);
C=confusionmat(y_test,y_test_hat);
acc=sum(diag(C))/sum(sum(C));
disp('Confusion Matrix using only color features:')
```

```
Confusion Matrix using only color features:
```

```
disp(C)
```

```
     7     0
     2     4
```

```
disp('Accuracy using only color features (in percent):')
```

```
Accuracy using only color features (in percent):
```

```
disp(acc*100)
```

```
       84.615
```

```

Fn_train=F1(:,31:40);
```

```
Fn_test=F2(:,31:40);
clf=fitcecoc(Fn_train,y_train);
y_test_hat=clf.predict(Fn_test);
C=confusionmat(y_test,y_test_hat);
acc=sum(diag(C))/sum(sum(C));
disp('Confusion Matrix using only boundary features:')
```

```
Confusion Matrix using only boundary features:
```

```
disp(C)
```

```
     4     3
     3     3
```

```
disp('Accuracy using only boundary features (in percent):')
```

```
Accuracy using only boundary features (in percent):
```

```
disp(acc*100)
```

```
       53.846
```

```

Fn_train=F1(:,41:58);
Fn_test=F2(:,41:58);
clf=fitcecoc(Fn_train,y_train);
y_test_hat=clf.predict(Fn_test);
C=confusionmat(y_test,y_test_hat);
acc=sum(diag(C))/sum(sum(C));
disp('Confusion Matrix using only region features:')
```

```
Confusion Matrix using only region features:
```

```
disp(C)
```

```
     7     0
     0     6
```

```
disp('Accuracy using only region features (in percent):')
```

```
Accuracy using only region features (in percent):
```

```
disp(acc*100)
```

```
   100
```

```

Fn_train=F1(:,59:end);
Fn_test=F2(:,59:end);
clf=fitcecoc(Fn_train,y_train);
y_test_hat=clf.predict(Fn_test);
C=confusionmat(y_test,y_test_hat);
acc=sum(diag(C))/sum(sum(C));
disp('Confusion Matrix using only texture features:')
```

```
Confusion Matrix using only texture features:
```

```
disp(C)
```

```
     7     0
     1     5
```

```
disp('Accuracy using only texture features (in percent):')
```

```
Accuracy using only texture features (in percent):
```

```
disp(acc*100)
```

```
92.308
```

Using all the features result in 100 percent accurate classification. According to accuracies, most important features would be region features. texture and color are the next two important features. Importance of features is different from last part which makes sense.

v)

```
CD=cd;
categories=dir('101_ObjectCategories');
categories(1:2)=[];
F1=[];
y1=[];
F2=[];
y2=[];
for i=1:numel(categories)
    cd([cd '\Features\' categories(i).name '\'])
    load(['features_' categories(i).name])
    F1=[F1;Fn_train];
    F2=[F2;Fn_test];
    y1=[y1;y_train];
    y2=[y2;y_test];
    cd(CD)
end
Fn_train=F1;
Fn_test=F2;
y_train=y1;
y_test=y2;
clf=fitcecoc(Fn_train,y_train);
y_test_hat=clf.predict(Fn_test);
C=confusionmat(y_test,y_test_hat);
acc=sum(diag(C))/sum(sum(C));
% commands for printing C are turned into comments because it is very big and takes so much space
% disp('Confusion Matrix using all features:')
% disp(C)
disp('Accuracy using all features (in percent):')
```

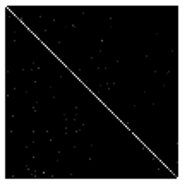
```
Accuracy using all features (in percent):
```

```
disp(acc*100)
```

```
89.243
```

```
%Normalizing C
C=C./repmat(sum(C,2),1,size(C,2));
C=C/max(C(:));
figure,
imshow(C)
title('Confiusion Matrix Visualization')
```

Confiusion Matrix Visualization



40 categories have 100 percent accuracy including "Faces", "bass", "brain", "cat_side", etc. The worst accuracy belongs to

"platypus" (zero percent). Four test platypus pictures are classified as motorbikes, butterfly, cougar_body and flamingo. Next worst category is "wrench".

Confidence (number 0 to 100% about your confidence in your performance on this project and optional statement about areas where you are particularly confidence or not):

100

Difficulty (number 0 to 100%):

90

Time Spent (hours):

~12 hours