

قسمت دوم

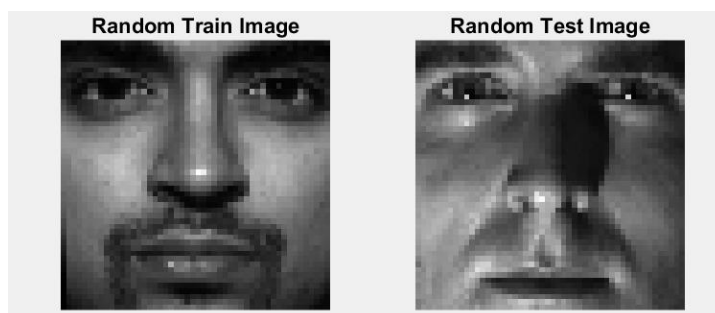
در این قسمت شما روش تصویر ویژه Eigenface را برای شناسایی چهره انسان پیاده‌سازی خواهید کرد. تصاویر چهره از Yale Face Database B بوده که در آن 64 تحت شرایط نور متفاوت از هر یک، جمعا 640، وجود دارد. با پیاده‌سازی شما قدرت تجزیه مقدار منفرد SVD را در نمایش تصاویر درخواهید یافت

الف.

پس از این که faces.zip را unzip کردید، پوشه‌ای را می‌یابید که حاوی همه تصاویر آموزش و تست است؛ train.txt و test.txt به ترتیب، مجموعه آموزش و تست را مشخص می‌کند، هر خطی مسیر تصویر و برچسب متناظر آن را ارائه می‌دهد

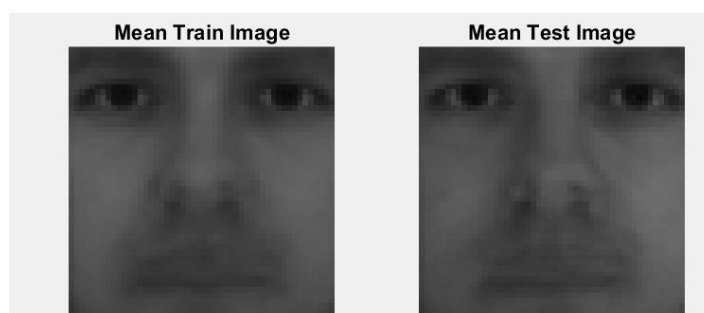
ب.

تصاویر مجموعه آموزش را در یک ماتریس X بارگذاری کنید: در مجموع 540 تصویر آموزش وجود دارد که هرکدام 50×50 پیکسل است و باید به یک بردار 2500 بعدی تبدیل شود؛ بنابراین سائز X باید 2500×540 باشد که هر سطری یک تصویر چهره تخت شده است. یک تصویر چهره از ماتریس X را انتخاب کرده و آن را نمایش دهید. مراحل گفته شده را برای مجموعه تست نیز انجام دهید. سائز ماتریس X_{test} برای مجموعه تست باید 100×2500 باشد



ج.

تصویر میانگین: تصویر میانگین μ از با استفاده از همه مجموعه آموزش محاسبه کنید. تصویر میانگین را نمایش دهید

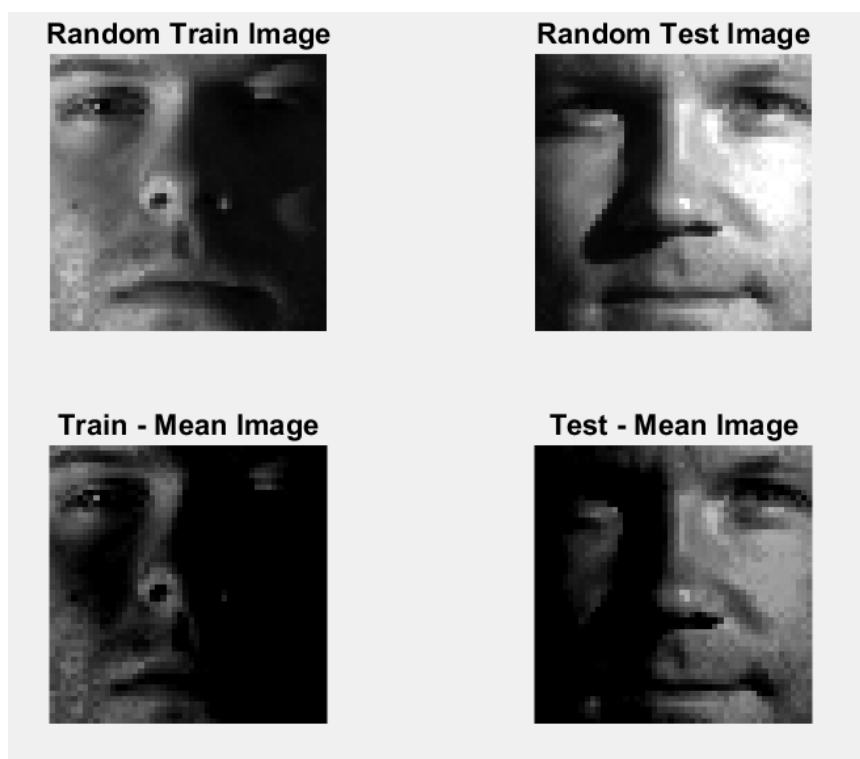


۵.

تفریق میانگین: تصویر میانگین را از همه سطرهای ماتریس X کم کنید. یک تصویر چهره را بعد تفریق میانگین انتخاب کرده و آن را نمایش دهید. همین کار را با استفاده از بردار میانگین محاسبه شده توسط تصاویر آموزش در قسمت (ج) برای تصاویر تست انجام دهید

سطر اول تصاویر اصلی و سطر دوم تصاویری است که میانگین از آنها کم شده است.

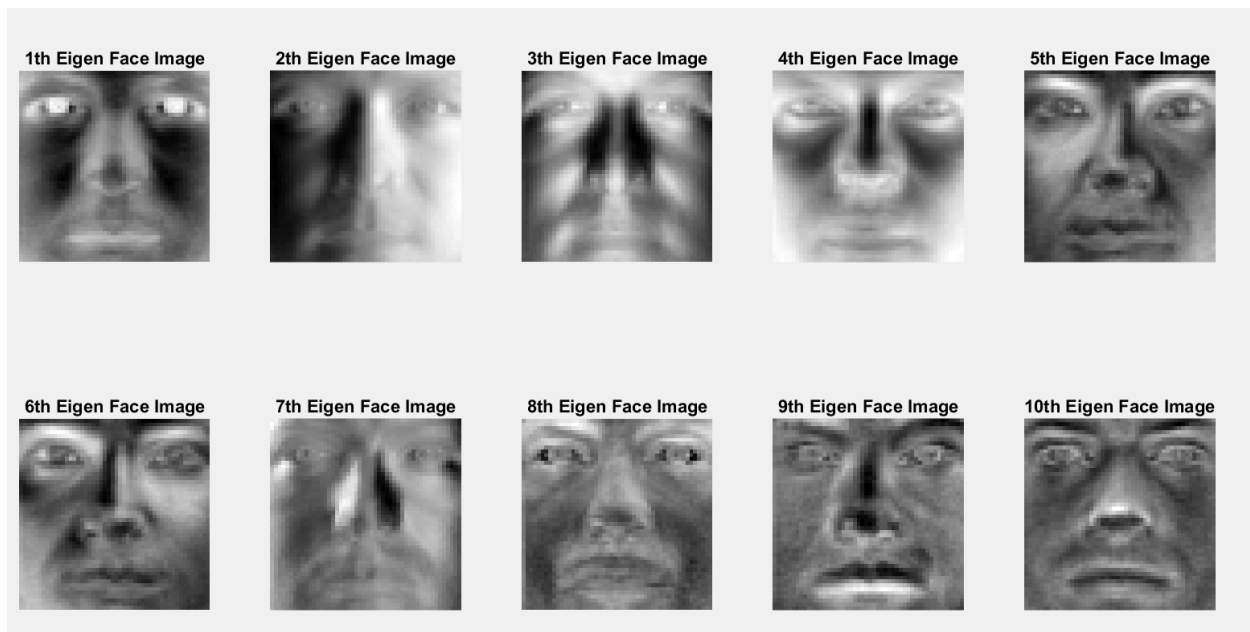
ستون اول مربوط به یک داده تصادفی از دادگان آموزش و ستون دوم مربوط به یک داده تصادفی از دادگان آزمایش



۵.

تصویر ویژه: تجزیه مقدار منفرد را روی مجموعه X ($X = U\Sigma V^T$) اجرا کنید تا ماتریس V^T حاصل شود، که هر سطر ماتریس V^T همان بعد تصاویر چهره را دارد. ما V_i (i امین سطر V^T) را به عنوان i امین تصویر ویژه در نظر می گیریم . 10 تصویر ویژه اول را نمایش دهید.

برای این کار از دستور svd متلب استفاده شده است. خروجی این دستور شامل ماتریس های U و V و Σ است. شکل زیر ده تصویر ویژه بردار V را نشان میدهد.



۹

تقریب مرتبه پایین: از آن جا که Σ یک ماتریس قطری با ترتیب غیرافزایشی است، می‌توانیم r عنصر اول Σ را با r ستون اول U و r سطر اول V^T برای تقریب \hat{X}_r استفاده کنیم. ماتریس \hat{X}_r تقریب مرتبه r ماتریس X نامیده می‌شود. نمودار خطای تقریب مرتبه r را به عنوان تابعی از r رسم کنید. $r=1, 2, 3, \dots, 200$

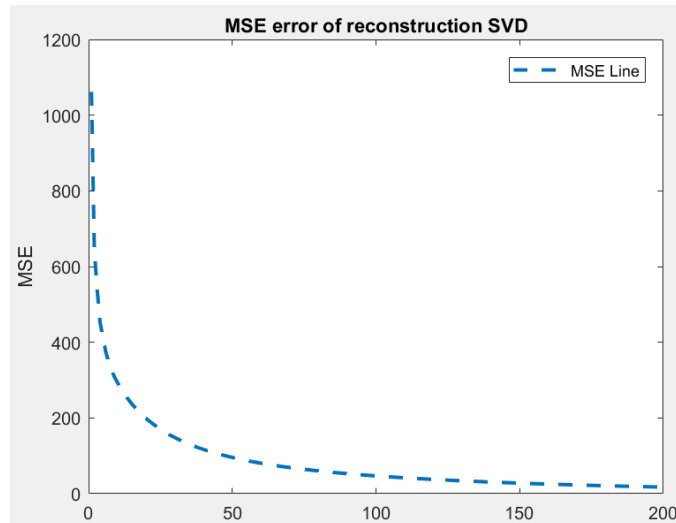
میدانیم که ماتریس X را میتوان با استفاده از ضرب سه مولفه حاصل از svd بدست آورد

$$X = U \Sigma V^T$$

اما میتوان از تمامی ظرفیت این سه ماتریس استفاده نکرد و فقط از r عنصر (سطر یا ستون) هر کدام بهره گرفت تا داده های اصلی را بازسازی کنیم

$$\hat{X} = U_r \Sigma_r V_r^T$$

دلیل استفاده از تنها r عنصر این است که بتوانیم مسئله را به ابعاد پایین تر ببریم تا سرعت و حتی دقت محاسبات را بالاتر ببریم و تنها از ویژگی هایی استفاده کنیم که بیشترین اطلاعات را به ما میدهند.



با توجه به نمودار بالا، متوجه میشویم که با بالاتر رفتن ابعاد انتخابی (مقدار r)، خطای بازسازی کمتر میشود و در نهایت در نقطه ای مثل 200، مشاهده میکنیم که خطای بازسازی نزدیک به صفر است. چنین نتیجه ای میتوان گرفت که با 200 بردار ویژه از تجزیه منفرد میتوان اصل داده ها را بازسازی کرد و نیاز نیست که در امر یادگیری مدل های آموزشی و دسته بندی، تمامی 2500 ویژگی را دخیل کرد. در نتیجه تنها میتوان از عناصر اول ماتریس های U و V و Σ استفاده کرد. این موضوع که داده ها را میتوان با تعداد عناصر مهم تجزیه منفرد svd بازسازی کرد، ما را به سمت کاهش ابعاد مسئله سوق میدهد، ابعادی که بیشترین اطلاعات را از دادگان به ما بدهد.

ز.

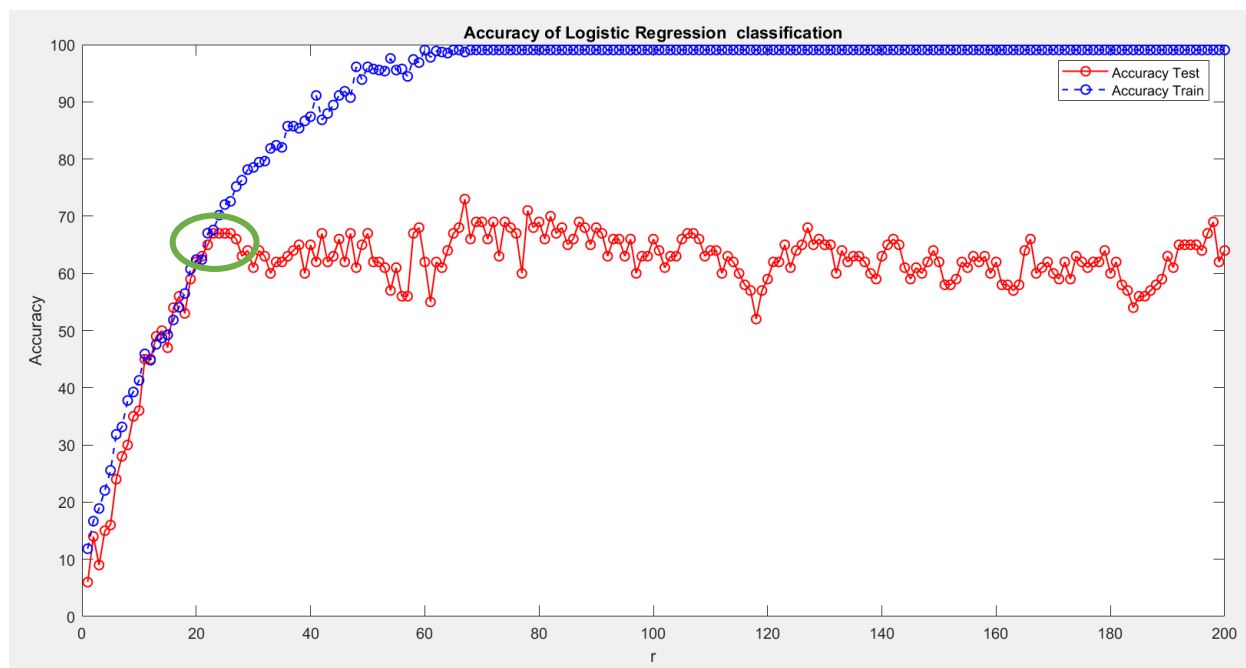
ویژگی های تصویر ویژه: r تصویر ویژه بالای تصاویر ویژه یک زیر فضای خطی r بعدی فضای تصاویر اصلی به نام فضای چهره را اسپن می کند که مرکز آن تصویر میانگین و محورهای آن تصاویر ویژه $\{V_1, V_2, \dots, V_r\}$ هستند؛ بنابراین با استفاده از r تصویر ویژه بالا می توانیم یک تصویر چهره z ، 2500 بعدی را به عنوان یک بردار ویژگی r بعدی $F \rightarrow V^T$ نمایش دهیم. تابعی بنویسید که ماتریس ویژگی r بعدی F و F_{test} را به ترتیب برای تصاویر آموزش X و تصاویر تست X_{test} تولید کند. (برای تولید F ماتریس X را در ترانهاده r سطر اول V^T ضرب کنید، F باید همان تعداد سطر X و r ستون داشته باشد؛ به طور مشابه برای X_{test})

تابع پیاده سازی شده	
ورودیها	
1. داده ها	
2. بردار ویژه برای تصویر داده ها به فضای جدید	
3. مقدار r که نشان دهد چه تعداد از بردار ویژه را انتخاب کنیم (داده ها در فضای جدید به چه ابعادی تصویر شوند)	
<pre>function F = Dimen_Reduction_Func(Data,Vec,r) F = Data * Vec(:,1:r); end</pre>	
خروجی نیز شامل داده های تصویر شده است	

ح.

شناسایی تصویر: ویژگی‌های آموزش و تست را به ازای $r=10$ استخراج کنید. یک مدل رگرسیون منطقی با استفاده از F آموزش دهید و روی F_{test} امتحان کنید. دقت طبقه‌بندی را روی مجموعه گزارش دهید. نمودار دقت طبقه‌بندی روی مجموعه تست را به عنوان تابعی از r رسم کنید. $r=1, 2, \dots, 200$

برای این قسمت از رگرسیون منطقی استفاده کرده ایم تا عملیات دسته بندی را انجام دهیم



نتیجه : در نمودار بالا، علاوه بر دقت دسته بندی روی داده‌های آموزشی، دقت دسته بندی را انجام داده ایم تا دقت آموزش مدل را نیز داشته باشیم. همانگونه که از نمودار مشخص است، دقت آموزش به مرور زیاد می‌شود و با افزایش مقدار r ، دقت آموزش نیز افزایش می‌یابد که نهایتاً به 100 درصد می‌رسد.

اما در طرف مقابل، دقت تست را بررسی می‌کنیم. دقت تست با افزایش r ، افزایش می‌یابد اما بعد از مقدار مشخصی از r ، دیگر افزایش دقت نداریم. این موضوع را میتوان به دلیل اشباع آموزش معرفی کرد. یعنی جایی که دقت آموزش بالاتر نمی‌رود و ثابت است و آموزش بیشتر و یا افزایش ابعاد، تاثیری در آموزش ندارد. و متعاقباً دقت تست نیز ثابت می‌ماند.

بهترین مقدار برای r را با ناحیه سبز رنگی نمایش داده ایم. جایی که دقت دسته بندی روی داده‌های آموزشی، ثابت می‌ماند. این مقدار حدود 20 است و افزایش ابعاد مسئله، تاثیری در افزایش دقت مدل ندارد. پس چنین میتوان گفت که از 2500 ویژگی یک عکس، تنها میتوان با حدود 20 ویژگی حاصل از تجزیه مقدار منفرد، به یک مدل مناسب دست یافت.

Run_me_2

```
%% Clear workspace
clc;
clear;
close all;

%% Read Train images part(a)
مشخص میکنیم که در کدام پوشه قرار داریم
thispath = pwd();

خواندن داده های آموزشی
File = fopen([thispath '\faces\train.txt'],'r');
count_tr = 0;
while 1
    tline = fgetl(File);
    if ~ischar(tline)
        break;
    end
    tline = strsplit(tline);

    خواندن عکس
    im = imread(tline{1});
    count_tr = count_tr + 1;

    قرار دادن عکس در متغیر X
    X(count_tr,:) = im(:)';

    مشخص کردن برچسب عکس خوانده شده
    Y(count_tr) = str2double(tline{2});
end
fclose(File);

خواندن داده های تست
%% Read Test images part(a)
File = fopen([thispath '\faces\test.txt'],'r');
count_ts = 0;
while 1
    tline = fgetl(File);
    if ~ischar(tline)
        break;
    end
    tline = strsplit(tline);
    im = imread(tline{1});
    count_ts = count_ts + 1;
    X_Test(count_ts,:) = im(:)';
    Y_Test(count_ts) = str2double(tline{2});
end
fclose(File);

نمایش یک عکس تصادفی از داده های آموزشی و تست
%% Show random image part(b)
figure;
Rnd_Idx = randi(count_tr);
subplot(1,2,1);
imshow(reshape(X(Rnd_Idx,:),50,50));
title('Random Train Image');
subplot(1,2,2);
Rnd_Idx = randi(count_ts);
imshow(reshape(X_Test(Rnd_Idx,:),50,50));
title('Random Test Image');
pause(0.1);
```

نمایش عکس میانگین

```
%% Show mean image of train and test data part(je)
figure;
subplot(1,2,1);
X_tr_Mean = uint8(mean(X));
imshow(reshape(X_tr_Mean,50,50));
title('Mean Train Image');
subplot(1,2,2);
X_ts_Mean = uint8(mean(X_Test));
imshow(reshape(X_ts_Mean,50,50));
title('Mean Test Image');
pause(0.1);
```

نرمال کردن داده ها = کم کردن مقدار عکس میانگین از عکس ها

```
%% normalize data part(d)
X_Train_Sub_Mean = bsxfun(@minus,X,X_tr_Mean);
X_Test_Sub_Mean = bsxfun(@minus,X_Test,X_tr_Mean);
```

نمایش عکسی تصادفی از عکس های نرمال شده

```
%% Show random image from normaize image part(d)
figure;
Rnd_Idx = randi(count_tr);
subplot(2,2,1);
imshow(reshape(X(Rnd_Idx,:),50,50));
title('Random Train Image');
subplot(2,2,3);
imshow(reshape(X_Train_Sub_Mean(Rnd_Idx,:),50,50));
title('Train - Mean Image');

Rnd_Idx = randi(count_ts);
subplot(2,2,2);
imshow(reshape(X_Test(Rnd_Idx,:),50,50));
title('Random Test Image');
subplot(2,2,4);
imshow(reshape(X_Test_Sub_Mean(Rnd_Idx,:),50,50));
title('Test - Mean Image');
pause(0.1);
```

محاسبه مقدار تجزیه منفرد

```
%% make SVD and show 10 first image from V part(ho)
A = double(X_Train_Sub_Mean);
[U,S,V] = svd(A);
```

نمایش ده تصویر ویژه اول

```
figure;
for i = 1 : 10
    subplot(2,5,i);
    imshow(reshape(V(:,i),50,50),[]);
    title([num2str(i) 'th Eigen Face Image']);
end
pause(0.1);
```

بازسازی داده ها با عناصر تجزیه منفرد ولی با r عنصر

```
%% reconstruct data using SVD part(v)
for r = 1 : 200
    Xr = U(:,1:r)*S(1:r,1:r)*V(1:r,:);
    error(r) = mse(A,Xr);
```

```

end
figure;
plot(1:r,error,'--','LineWidth',2);
title('MSE error of reconstruction SVD');
xlabel('r');
ylabel('MSE');
legend('MSE Line');
pause(0.1);

تشخیص چهره با استفاده از روش رگرسیون منطقی
%% Face recognition with SVM part(z)
آرایه ای برای ذخیره مقادیر دقت دسته بندی

Acc_arr_Tr = [];
Acc_arr = [];

for r = 1 : 50
    fprintf('Face recognition with %d eigen face\n',r);

    کاهش ابعاد یا همان تصویر کردن داده با r مقدار بردار ویژه
    F = Dimen_Reduction_Func(double(X_Train_Sub_Mean),V',r);
    F_Test = Dimen_Reduction_Func(double(X_Test_Sub_Mean),V',r);

    آموزش رگرسیون با دستور mnrfiit
    % Learn Logistic Regression
    [B,dev,stats] = mnrfiit(F,Y','model','hierarchical');

    تست مدل بدست آمده با استفاده از دستور mnrrval روی دادگان آموزشی
    % Test on train images
    P = mnrrval(B,F,'model','hierarchical');
    [~, Y_Test_Pred] = max(P,[],2);
    Acc_arr_Tr(r) = mean(Y==Y_Test_Pred')*100;

    تست مدل بدست آمده با استفاده از دستور mnrrval روی دادگان تست
    % Test on test images
    P = mnrrval(B,F_Test,'model','hierarchical');
    [~, Y_Test_Pred] = max(P,[],2);
    Acc_arr(r) = mean(Y_Test==Y_Test_Pred')*100;
end

نمایش خطای حاصل

figure;
plot(1:r,Acc_arr,'r-o','LineWidth',1);
hold on;
plot(1:r,Acc_arr_Tr,'b--o','LineWidth',1);
title('Accuracy of Logistic Regression classification');
xlabel('r');
ylabel('Accuracy');
legend('Accuracy Test','Accuracy Train');
pause(0.1);

```