(باسمه تعالی)

گزارش پروژه ی پایانترم

جبرخطی

کاری از:

مهدی رحمانی

9731701

فهرست

چکیده

در این پروژه می توان به طور کلی گفت که با مفاهیمی اولیه از پردازش تصویر آشنا شدیم. روند کلی کار به صورت خلاصه در ادامه آمده است:

در ابتدای کار عکسی را تبدیل به ماتریس کرده و حالا ما یک سری دیتا داریم که میتوان روی آنها عملیات های مختلفی انجام داد.

این ماتریس در ابتدا یک آرایه تصویری از نوع داده‌ای (uint8) است و اعداد صحیح ذخیره شده مقادیری بین 0 و 255دارند.

در مرحله ی بعد با تبدیل این ماتریس به دابل ، دیتاها مقادیری بین 0 و 1 میابند هر مقداری که بین 0 و 1 قرار داشته باشد، با رنگ سطح خاکستری (grayscale) نمایش داده می‌شود. مقادیر بزرگتر از 1، با رنگ سفید و مقادیر کوچکتر از 0 نیز با رنگ سیاه نمایش داده خواهند شد .

طبق آنچه که در کتاب آمده است میتوان از این ماتریس به یک ماتریس با 3 سطر که هرسطر دارای دیتاها و پیکسل های مربوط به یکی از رنگ های قرمز و سبز و آبی (RGB) میباشد؛ رسید.

هرستون این ماتریس مانند یک بردار است به 3 مولفه و لذا میتوان آن ها را دریک پلات نمایش داد که ابر داده ها گفته میشد که در کتاب هم با عنوان اسکاتر پلات آورده شده است.

در مرحله بعد ازما خواسته شده تا میانگین داده ها را بیابیم که طبق روش گفته شده در کتاب تمامی ستون ها رو با یکدیگر جمع کرده و بر تعداد پیکسل ها تقسیم میکنیم که در نهایت یک بردار با 3 مولفه به ما میدهد.

سپس ماتریس PCA یا همان ماتریس کواریانس را با کمک میانگینی که در مرحله قبل یافتیم و روش گفته شده در کتاب میابیم.البته می توان از دستورcov در متلب هم برای به دست آوردن آن استفاده کرد.

حال باتوجه به آنچه که در کتاب آورده شده مجموع عناصر روی قطر اصلی ماتریس کواریانس برابر با واریانس داده ها می باشد.هم چنین با دستوری در متلب می توان به ماتریس همبستگی دست یافت. این ماتریس که یک ماتریس متقارن با عناصر 1 روی قطر اصلی میباشد میتواند اطلاعاتی راجع به میزان وابستگی مشخصه های مورد بررسی (که در این جا رنگ های قرمز و آبی وسبز بودند) به ما بدهد. به کمک واریانس و همبستگی داده ها می توان تحلیل های مفیدی روی داده ها داشت.

همانطور که پیشتر اشاره شد داده ی ما 3 بعدی میباشد . میتوان بعد آن را به کمک روش واریانس کل داده ها کم کرد منتها با کاهش بعد مقداری از دیتا ها هم از دست میرود.

در نهایت با استفاده از روش principle component analyze یک عکس دیگر با اعمال تغییرات و عملیاتی روی داده های اولیه تولید میکنیم.

می توان از این مجموعه کارها این برداشت را کرد که عکس ها میتوانند بسیار پیچیده تر از آنچیزی باشند که در نگاه اول دیده میشوند. مجموعه ای از دیتا ها هستند که میتوان اطلاعات مختلفی از آن ها استخراج کرد و کارهای بسیار خارق العاده ای روی آن ها انجام داد. میتوان تاثیر سه هرکدام از این 3 دسته از رنگ های اصلی را بر یک عکس مشاهده کرد و درنهایت گره خوردن ریاضیات وآمار و جبر را با این پردازش ها مشاهده کرد.

مقدمه

همانطور که گفته شد با مفاهیم اولیه پردازش تصویر در این پروژه آشنا میشویم و میتوان گفت :

پردازش تصویر به مجموعه‌ای از تکنیک‌هایی اطلاق می‌شود که با هدف تبدیل یک تصویر به قالب دیجیتال و انجام اعمال محاسباتی بر روی آن شکل گرفته‌اند. هدف از انجام اعمال محاسباتی مرتبط با پردازش تصویر در متلب، تولید نسخه‌ای بهبود یافته از تصاویر دیجیتالی و یا استخراج اطلاعات با معنی و مفید از آن‌ها است.

پروژه ی مذکور در محیط برنامه نویسی متلب انجام گرفته لذا توابع و ابزارهایی که بررسی میکنیم مربوط به این محیط میباشد.

برای انجام عملیات محاسباتی متناظر با پردازش تصویر در متلب، ابتدا باید تصاویر دیجیتال از طریق واسط‌هایی نظیر اسکنر نوری و دوربین‌های دیجیتال تولید شوند. سپس، تصاویر دیجیتال تولید شده تحلیل می‌شوند. در مرحله بعد، تصاویر دیجیتالی از طریق فرآیندهایی نظیر فشرده‌سازی داده‌ها ، بهبود تصاویر، فیلتر تصاویر و سایر موارد، مورد دستکاری عددی قرار گرفته و در نهایت، تصاویر خروجی مطلوب تولید می‌شوند.

در این محیط برنامه‌نویسی، توسعه‌دهندگان قادرند تا از روش‌های دستکاری ماتریسی، توابع، روش‌های نمایش داده و الگوریتم‌های توسعه داده شده در متلب، برای پیاده‌سازی روش‌های هوشمند (مبتنی بر نمایش عددی و ماتریسی) دلخواه خود استفاده کنند.

الگوریتم‌های پردازش تصویر در متلب، مجموعه‌ای از توابع هستند که قابلیت‌های محیط محاسبات عددی متلب را گسترش می‌دهند. تولباکس پردازش تصویر در متلب، مجموعه‌ای از «الگوریتم‌های مرجع استاندارد» را برای کاربردهای پردازش، تحلیل و نمایش بصری تصاویر و همچنین توسعه الگوریتم‌های پردازش تصویر در متلب فراهم می‌آورد

از الگوریتم‌های پردازش تصویر در متلب، می‌توان برای بخش‌بندی تصاویر، بهبود تصاویر، کاهش نویزدر تصاویر، تبدیلات هندسی، انطباق تصویرو انجام عملیات پردازش تصویر 3-بعدی استفاده کرد.

انواع تصاویر

در تولباکس پردازش تصویر در متلب و توابع موجود در آن، چهار نوع تصویر پشتیبانی می‌شود:

- تصاویر سطح خاکستری(Grayscale | Grey-level Images)
- تصاویر باینری(Binary Images)

- تصاویر شاخص‌گذاری شده(Indexed Images)
- تصاویرRGB

تصاویر سطح خاکستری

در این دسته از تصاویر، که به آن‌ها تصاویر «تک‌رنگ» (Monochrome) نیز گفته می‌شود، از 8 بیت برای نمایش مقدار شدت رنگ هر پیکسل استفاده می‌شود؛ پیکسل با مقدار شدت برابر با صفر، رنگ سیاه را نمایش می‌دهد. همچنین، پیکسل با مقدار شدت برابر با 255، رنگ سفید خواهد بود. در نهایت، پیکسل‌هایی که مقادیری بین 0 و 255 دارند، طیف‌های خاکستری را نمایش می‌دهند. تصاویر سطح خاکستری، توسط آرایه‌های دوبعدی و مقادیر پیکسل‌ها نیز، توسط یک عدد 8 بیتی نمایش داده می‌شوند.

تصاویر باینری

در این دسته از تصاویر، از تنها 1 بیت برای نمایش مقدار پیکسل‌ها استفاده می‌شود؛ مقدار 1، به معنای رنگ سفید و مقدار 0، به معنای رنگ سیاه. تصاویر سیاه و سفید نیز توسط آرایه‌های دوبعدی نمایش داده می‌شوند. حجم کم این دسته از تصاویر، مهم‌ترین مزیت آن‌ها محسوب می‌شود.

تصاویر شاخص‌گذاری شده

این تصاویر، ماتریسی از مقادیر صحیح هستند. در این ماتریس، هر مقدار صحیح به یک سطر خاص از مقادیرRGB ، در یک ماتریس نقشه ثانویه به نام «نقشه رنگ» (Colour Map) اشاره دارد.

تصاویرRGB

در تصاویرRGB ، هر پیکسل رنگی توسط یک سه‌تایی مشخص کننده مؤلفه‌های قرمز(R) ، سبز (G) و آبی (B) آن پیکسل، نمایش داده می‌شود. در تولباکس پردازش تصویر در متلب و توابع آن، هر تصویر رنگی RGB متناظر با یک آرایه سه‌بعدی به ابعاد MxNx3 است. در اینجا، Mبیانگر ارتفاع وN ، نشان دهنده پهنای تصویر است. همچنین، عدد 3 بیانگر تعداد مؤلفه‌های رنگی تصویر RGB است. در تصاویر RGB از نوعdouble ، محدوده مقادیر شدت پیکسل‌ها، بازه [0 و 1]

است. در تصویر RGB از نوع Unit8 وUnit16 ، محدوده مقادیر به ترتیب برابر با بازه [0 و 255] و [0 و 65535] است.

شایان توجه است که بیشتر الگوریتم‌های پردازش تصویر در متلب (از نوع پردازش تصاویر تک‌رنگ)، از تصاویر باینری یا سطح خاکستری برای انجام عملیات پردازشی استفاده می‌کنند

برخی از توابع استفاده شده به صورت مختصر توضیح داده میشوند:

Imread: imread('filename')

برای خواندن تصاویر در محیط متلب، از تابع (imread) استفاده می‌شود. شایان توجه است زمانی که آدرس فیزیکی محل ذخیره‌سازی تصویر، در آرگومان (filename) قید نشده باشد، تابع (imread) تصویر را از دایرکتور کنونی متلب خواهد خواند. در شرایطی که نیاز باشد تا تصویر از دایرکتوری دیگری در متلب خوانده شود، لازم است تا آدرس کامل تصویر در آرگومان (filename) مشخص شود.

Im2double: im2double(I)

دستور im2double در متلب، تصویر را به یک تصویر از نوع دابل (double) تبدیل می‌کند. این دستور در پردازش تصویر بسیار پرکاربرد است. تصاویری که با استفاده از دستور imread در متلب فراخوانی می‌شوند، از نوع uint8 هستند. با استفاده از دستور im2double عمل تبدیل به double را به‌راحتی می‌توان انجام داد.

هم جنین توابعی مانند cov(x) یا mean(x) یا corrcoef(x) به ترتیب برای به دست آوردن میانگین و کواریانس و همبستگی کاربرد دارند.

توضیحات اصلی گزارش

مرحله اول: یک عکس از ناحیه ی موردنظر خود بگیرید(تصویرخودتان یا یک منظره).

عکسی که من انتخاب کردم عکس زیر می باشد:

اسم فایل pic1.jpg می باشد.

مرحله دوم: آن را به صورت ماتریس ذخیره کنید:

ابتدا به کمک دستور pic_matrix=imread('pic1.jpg') عکس مورد نظر را خوانده وبه صورت یک ماتریس در pic_matrix ذخیره میکنیم که یک ماتریس با ابعاد 3*300*250 می باشد و مقادیر اعداد صحیح ذخیره شده در متغیرpic_matrix را به مقادیری بین 0 و 255 محدود می‌کند .

سپس به کمک rgb=im2double(pic_matrix) ماتریس داده های به دست آمده را در rgb به صورت دابل ذخیره کرده که در این حالت مقادیر ذخیره شده بین 0 و 1 می باشند.

مرحله سوم:ابرداده های آن را رسم کنید.

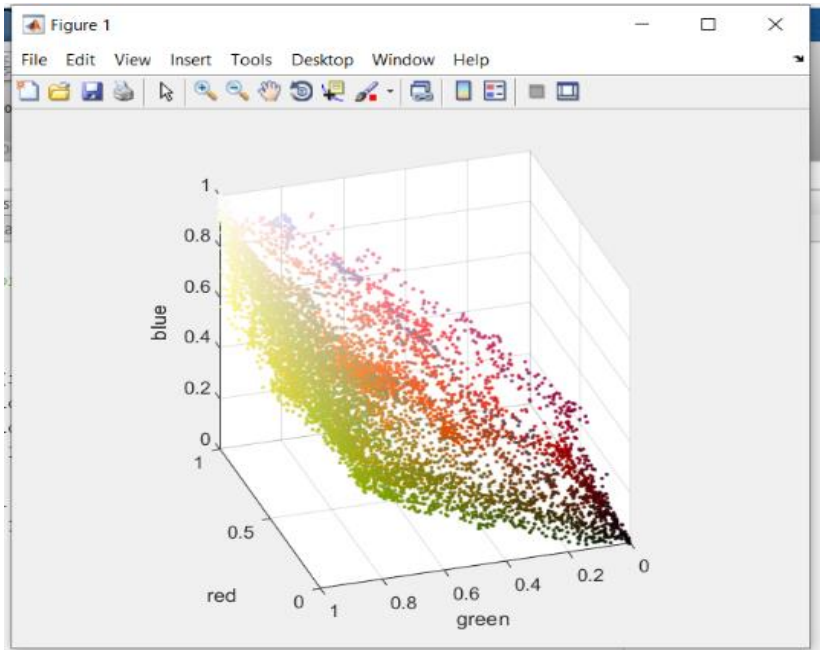
برای رسم داده ها نیاز داریم که ماتریسی داشته باشیم که دارای 3 سطر (که هرکدام بیانگر یک مشخصه قرمز و سبز و آبی می باشد) و 250*300 ستون(به عداد کل پیکسل ها) داشته باشد که در این حالت هر ستون بیانگر یک مختصات یک نقطه درR^3 میباشد. این کار به کمک rgb_2D=reshape(rgb,[],3) انجام شده.

سپس برای به دست آوردن تعداد ستون ها از n=size(rgb_2D,1) استفاده کردیم. از آنجایی مه تعداد داده ها بسیار زیاد است و ران کردن برنامه طولانی میشود و به مشکل میخورد ما 10 تا 10 تا دیتاها را پلات گرفتیم.(درصورت لزوم میتوان به شمارنده حلقه for رجوع کرده وi=1:10:n را بهi=1:n تغییر داد تا کل داده ها پلات شوند)

کد مربوط به این قسمت در تصویر زیر آمده است:

```
7    % Convert 3-dimensional array to 2D, where each row is a pixel (RGB)
8 -  rgb_2D = reshape(rgb, [], 3);
9    % n is the number of pixels:
10 - n = size(rgb_2D,1);
11   % Plot pixels in color space with limition of plot every 10th point
12 - figure(1)
13 - hold on
14 - for i=1:10:n
15 -     colour = rgb_2D(i,:);
16 -     colour = max(colour, [0 0 0]);
17 -     colour = min(colour, [1 1 1]);
18 -     plot3(rgb_2D(i, 1), rgb_2D(i, 2), rgb_2D(i, 3),'.', 'Color', colour);
19 - end
20 - xlabel('red'), ylabel('green'), zlabel('blue');
21 - xlim([0 1]), ylim([0 1]), zlim([0 1]);
22 - hold off
23 - grid on
24 - axis equal
25   %plot rotation.
26 - for az=-180:3:180
27 -     view(az,30);
28 -     drawnow;
29 - end
```

حال اگر برنامه ران کنیم نتیجه به صورت زیر خواهد بود(البته به خاطر کد موجود در خط های 26 تا 29 پلات (ابر داده ها) میچرخد تا بتوان از زوایای مختلف به صورت بهتر داده ها را در فضا دید.



مرحله 4: میانگین داده ها را بیابید.

دراین مرحله میانگین داده ها را باید بیابیم که طبق کتاب باید مجموع داده های مربوط به هر رنگ را بیابیم و سپس تقسیم بر تعداد آن ها بکنیم.

Mean and Covariance

To prepare for principal component analysis, let $[\,\mathbf{X}_1 \quad \cdots \quad \mathbf{X}_N\,]$ be a $p \times N$ matrix of observations, such as described above. The **sample mean**, $\mathbf{M}$, of the observation vectors $\mathbf{X}_1, \ldots, \mathbf{X}_N$ is given by

$$\mathbf{M} = \frac{1}{N}(\mathbf{X}_1 + \cdots + \mathbf{X}_N)$$

کد مربوط به این قسمت در عکس زیر آمده:

```
32      %%%% step4 %%%%
33      %finding mean
34 -    sum=zeros(1,3);
35 -    for i=1:n
36 -        sum(1,1)=sum(1,1)+rgb_2D(i, 1);
37 -        sum(1,2)=sum(1,2)+rgb_2D(i, 2);
38 -        sum(1,3)=sum(1,3)+rgb_2D(i, 3);
39 -    end
40 -    mean_of_data =(1/n)*(sum);
```

نتیجه کد بالا به صورت زیر میباشد:

mean_of_data

1x3 double

	1	2	3	4	5	6	7	8	9	10	11	12	13
1	0.7534	0.5815	0.3058										
2													
3													
4													
5													
6													
7													
8													
9													
10													
11													
12													
13													
14													
15													
16													
17													
18													
19													

مرحله 5: با استفاده از قسمت 4، ماتریس(covariance(PCA را بسازید.

طبق آنچه در کتاب آمده است عمل میکنیم و ماتریس های گفته شده را با توجه به میانگین به دست آمده در مرحله قبل میابیم و سپس آن را در فرمول گذاشته و جواب را حساب میکنیم.

For the data in Fig. 1, the sample mean is the point in the "center" of the scatter plot. For $k = 1, \ldots, N$, let

$$\hat{\mathbf{X}}_k = \mathbf{X}_k - \mathbf{M}$$

The columns of the $p \times N$ matrix

$$B = [\hat{\mathbf{X}}_1 \quad \hat{\mathbf{X}}_2 \quad \cdots \quad \hat{\mathbf{X}}_N]$$

have a zero sample mean, and B is said to be in **mean-deviation form**. When the sample mean is subtracted from the data in Fig. 1, the resulting scatter plot has the form in Fig. 3.

atrices and Quadratic Forms

The **(sample) covariance matrix** is the $p \times p$ matrix S defined by

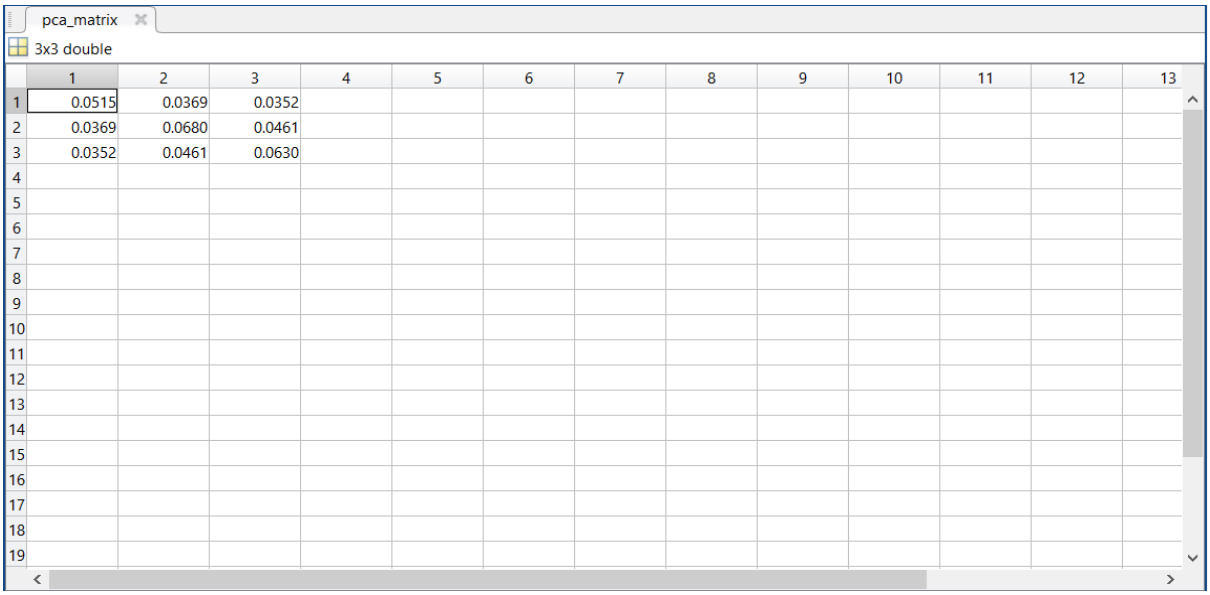
$$S = \frac{1}{N-1} BB^T$$

Since any matrix of the form BB^T is positive semidefinite, so is S. (See Exercise 25 in Section 7.2 with B and B^T interchanged.)

کد مربوط به این قسمت نیز در ادامه آمده است:

```
42
43      %%% step5 %%%
44      %finding PCA matrix
45 -    B=rgb_2D-mean_of_data;
46 -    pca_matrix=(1/(n-1))*(B')*(B);
47 -    disp('the PCA matrix is:'),disp(pca_matrix);
```

نتیجه کد بالا به صورت زیر میباشد:


pca_matrix

3x3 double

	1	2	3	4	5	6	7	8	9	10	11	12	13
1	0.0515	0.0369	0.0352										
2	0.0369	0.0680	0.0461										
3	0.0352	0.0461	0.0630										
4													
5													
6													
7													
8													
9													
10													
11													
12													
13													
14													
15													
16													
17													
18													
19													

مرحله 6: مقدار واریانس و همبستگی داده ها را محاسبه و براساس مقدار به دست آمده آن را تحلیل کنید.

طبق آنچه که درکتاب آورده شده واریانس داده برابر با مجموع عناصر روی قطر اصلی ماتریس کوواریانس می باشد.

The **total variance** of the data is the sum of the variances on the diagonal of S. In general, the sum of the diagonal entries of a square matrix S is called the **trace** of the matrix, written $\text{tr}(S)$. Thus

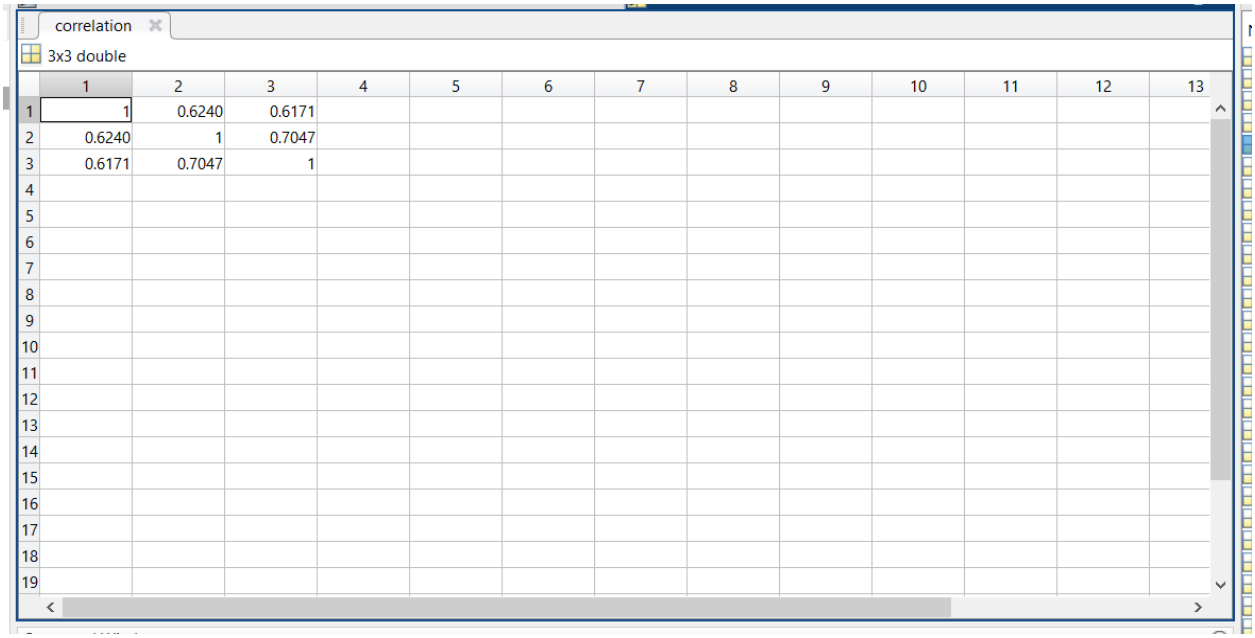
$$\{\text{total variance}\} = \text{tr}(S)$$

بنابراین می توان گفت:

Var=0.0515 + 0.0680+ 0.0630=0.1825

همانطور که دیده میشود واریانس داده عددی کوچک میباشد که به معنای این است که دیتا های مورد نظر خیلی پراکنده نمیباشند.

برای حساب کردن همبستگی داده ها می توان از corrcoef(rgb_2D) استفاده کرد که ماتریس همبستگی را به ما میدهد که به صورت زیر میباشد:



	1	2	3	4	5	6	7	8	9	10	11	12	13
1	1	0.6240	0.6171										
2	0.6240	1	0.7047										
3	0.6171	0.7047	1										
4													
5													
6													
7													
8													
9													
10													
11													
12													
13													
14													
15													
16													
17													
18													
19													

چیزی که میتوان به صورت کلی فهمید این است که اولا داده ها پراکندگی زیاد ندارند و هم چنین به هم وابسته اند و چون اعدادی که در خارج از قطر اصلی ماتریس همبستگی قرار دارند اعداد به نسبت بزرگی (در بازه 0 تا1) هستند پس با یکدیگر وابستگی مثبت دارند. پس یعنی اگر برای مثال هرچه میزان رنگ قرمزبیشتر باشد میزان رنگ سبز یا زرد هم بیشتر است.

کد مربوط به این قسمت در زیر آمده است:

```
48
49      %%% step6 %%%
50      % finding variance and correlation of data
51 -    var=pca_matrix(1,1)+pca_matrix(2,2)+pca_matrix(3,3);
52 -    disp('variance of data:'),disp(var);
53 -    correlation=corrcoef(rgb_2D);
54 -    disp('correlation of data:'),disp(correlation);
55
```

مرحله 7: **داده ای که درحال حاضر با آن کارمیکنید 3بعدی است. آیا بعد آن را میتوان کاهش داد؟ برای این منظور از روش واریانس کل داده ها استفاده کنید.**

بله میتوان بعد را کاهش داد اما خب بخشی از دیتا ها کم میشود. برای این کار در ماتریس PCA که ساختیم مقادیر ویژه و بردارهای ویژه را محاسبه میکنیم. سپس می توانیم آن را به صورت قطری متعامد در بیاوریم . در این حالت واریانس کل داده ها تغییر نمیکند. همانطور که در کتاب آمده است داریم:

Reducing the Dimension of Multivariate Data

Principal component analysis is potentially valuable for applications in which most of the variation, or dynamic range, in the data is due to variations in *only a few* of the new variables, $y_1, \ldots, y_p$.

It can be shown that an orthogonal change of variables, $\mathbf{X} = P\mathbf{Y}$, does not change the total variance of the data. (Roughly speaking, this is true because left-multiplication by P does not change the lengths of vectors or the angles between them. See Exercise 12.) This means that if $S = PDP^T$, then

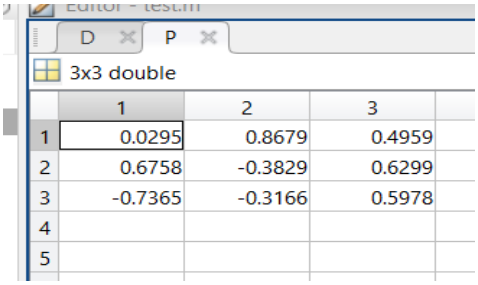
$$\begin{Bmatrix}\text{total variance}\\ \text{of } x_1, \ldots, x_p\end{Bmatrix} = \begin{Bmatrix}\text{total variance}\\ \text{of } y_1, \ldots, y_p\end{Bmatrix} = \text{tr}(D) = \lambda_1 + \cdots + \lambda_p$$

The variance of y_j is λ_j, and the quotient $\lambda_j / \text{tr}(S)$ measures the fraction of the total variance that is "explained" or "captured" by y_j.

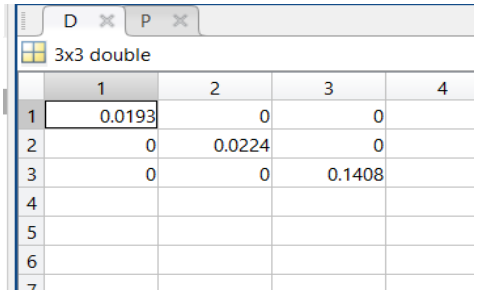
پس از انجام دادن این کار و به دست آوردن ماتریس های P وD سپس ماتریس A را طوری میسازیم که سطر اول آن ترانهاده بردار ویژه مربوط به بزرگ ترین مقدارویژه باشد و سطر دوم و سوم آن هم به ترتیب ترانهاده بردار ویژه های مربوط به دومین وسومین مقادیر ویژه باشد.

حال میخواهیم بردار های ورودی را داخل فضای تولید شده با بردارهای ویژه تصویر کنیم. برای این کار ابتدا میانگینی که در مرحله 4 به دست آوردیم را از هرستون ماتریس rgb_2D کم میکنیم. سپس ترانهاده ماتریس حاصل را از راست در ماتریس A ضرب میکنیم. ماتریس حاصل ماتریس Y میباشد که 3*n میباشد. حال اگر به ترتیب تصاویر مربوط به سطر اول و دوم و سوم را نشان دهیم خواهیم دید که تصویر مربوط به سطر اول که متناظر با بیشترین مقدار ویژه بود وضوح بیشتری دارد.

ماتریس P و D مربوط به عکس pic1.jpg در زیر آمده:(البته مقادیرویژه به صورت نزولی مرتب شده)



	1	2	3
1	0.0295	0.8679	0.4959
2	0.6758	-0.3829	0.6299
3	-0.7365	-0.3166	0.5978
4			
5			



	1	2	3	4
1	0.0193	0	0	
2	0	0.0224	0	
3	0	0	0.1408	
4				
5				
6				

tr(D)=0.0193+0.0224+0.1408=0.1825

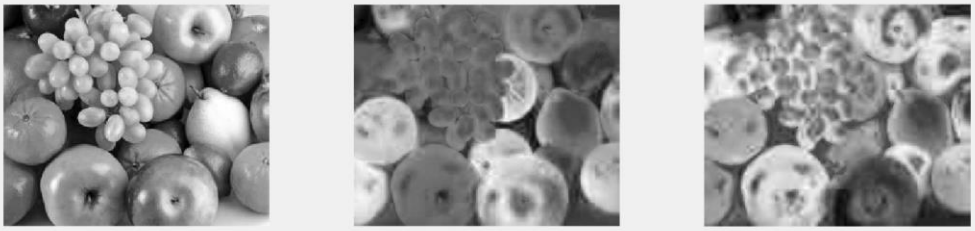
first component: 0.1408/0.1825=77.15%

second component:0.0224/0.1825=12.28%

third component:0.0193/0.1825=10.57%

همانطور که از محاسبات هم پیداس حدود 77 درصد اطلاعات توسط مشخصه اول منتقل میشود. و بنابراین اگر مثلا بردار سوم را حذف کنیم از تصویر اطلاعات کمی حذف میشود.

برای واضح بودن این موضوع هر بردار رو جداگانه به عکس تبدیل کرده و محتوا را نشان می دهیم:



همچنین کد مربوط به این قسمت عبارت است از:

```
%%% step 7%%%
% Get eigenvalues and eigenvectors of pca_matrix.
% Produces P,D such that pca_matrix*P = P*D.
% So the eigenvectors are the columns of P.
[P,D] = eig(pca_matrix);
e1 = P(:,3);
disp('Eigenvector e1:'), disp(e1);
e2 = P(:,2);
disp('Eigenvector e2:'), disp(e2);
e3 = P(:,1);
disp('Eigenvector e3:'), disp(e3);
d1 = D(3,3);
disp('Eigenvalue d1:'), disp(d1);
d2 = D(2,2);
disp('Eigenvalue d2:'), disp(d2);
d3 = D(1,1);
disp('Eigenvalue d3:'), disp(d3);

A = [e1'; e2'; e3'];
Y = A*(rgb_2D - repmat(mean_of_data,n,1))';
[height,width,depth] = size(rgb);
Y1 = reshape(Y(1,:), height, width);
Y2 = reshape(Y(2,:), height, width);
Y3 = reshape(Y(3,:), height, width);

figure(2);
subplot(1,3,1), imshow(Y1,[]);
subplot(1,3,2), imshow(Y2,[]);
subplot(1,3,3), imshow(Y3,[]);
```

مرحله 8:با استفاده از روش principle component analyze حداقل یک عکس جدید از ورودی خود تولید کنید.

حال میخواهیم به کمک روش **principle component analyze** یک عکس جدید درست کنیم. به این صورت که با همون دوبرداری که بیشترین دیتا درآن هاست یعنی Y1 و Y2 عکس رو دوباره میسازیم. طبیعتا یکسری دیتاها از دست میرود ولی خب دیتاهای کمی می باشد.

برای این کارمیانگین به دست آمده را با ضرب ترانهاده ماتریس A در y جمع میکنیم و ماتریس new_pic را تولید میکنیم که یک ماتریس n*3 می باشد.

A(1:2,:) به دوسطر اول ماتریسA و همچنین Y(1:2,:) نیز به دو سطر اول Y اشاره دارد.

کد مربوط به این قسمت در زیر آمده است:

```
90
91      %%% step 8 %%%
92      % Reconstruct image using only Y1 and Y2.
93 -    new_pic = ( A(1:2,:)' * Y(1:2,:) )' + repmat(mean_of_data,n,1);
94
95 -    Ir(:,:,1) = reshape(new_pic(:,1), height, width);
96 -    Ir(:,:,2) = reshape(new_pic(:,2), height, width);
97 -    Ir(:,:,3) = reshape(new_pic(:,3), height, width);
98 -    figure(3)
99 -    subplot(1,2,1),imshow(Ir);
100 -   subplot(1,2,2),imshow(pic_matrix);
101
102
```

درنهایت خروجی به صورت زیر میباشد. عکس سمت چپ به کمک PCA به دست آمده و عکس سمت راست عکس اولیه میباشد.

نتیجه گیری

پردازش تصویر در متلب و توابع موجود در آن، مجموعه وسیعی از روش‌ها، الگوریتم‌ها و تکنیک‌ها را شامل می‌شوند. فرایندهایی نظیرImage Sharpening ، حذف نویز، Deblurring، استخراج لبه‌ها، باینری‌سازی تصاویر، بهبود کنتراست تصاویر، بخش‌بندی و برچسب‌گذاری اشیاء در تصویر، از جمله مهم‌ترین فرایندهای بنیادی پردازش تصویر در متلب هستند که تقریبا در تمامی تکنیک‌ها و الگوریتم‌های پردازش تصویر مورد استفاده قرار می‌گیرند.

با کمک درس های جبر وآمار و ریاضیات و با ابزاری نظیر متلب میتوان اطلاعات مفید و دلخواهی را از یک عکس خارج کرد که در کنار جذابیت هایی که دارد میتوانند بسیار کاربردی باشد.

در این پروژه فهمیدیم که ماهیت تصاویری که در کامپیوتر ها مشاهده میکنیم دیتاهایی میباشد که میتوان در یک ماتریس آن ها را ذخیره کرد و درواقع پلی شد تا فهم آن مباحثی که در جبر خطی و آمار مطالعه شده؛ آسان تر شود و کاربرد آن ها به صورت عملی بهتر درک شود.