دانشگاه صنعتی امیرکبیر

(پلی تکنیک تهران)

دانشکده مهندسی کامپیوتر و فناوری اطلاعات

گزارش پروژه پایانی جبر خطی کاربردی

بررسی کاربرد های جبر خطی در پردازش تصویر

نگارش :

علی فرجی

استاد درس :

دکتر احسان ناظر فرد

بهمن ماه ۹۷

فهرست مطالب

۱ چهره های پایه برای شناسایی چهره

در این بخش ما ابتدا تمامی تصاویر را خوانده و در لیستی ذخیره کردیم سپس میانگین تصاویر را محاسبه کرده و از همه تصاویر آن را کم میکنیم و در لیست Q می ریزم.

حال برای محاسبه ماتریس کوواریانس هر عضو Q را در ترانهاده اش ضرب کرده همه را با هم جمع میکنیم و در آخر بر تعداد Q ها که همان تعداد تصاویر است تقسیم میکنیم.

حال ۶ تا از بزرگترین مقادیر ویژه و بردار های ویژه متناظر آنها را برای ماتریس کوواریانس پیدا کرده و حال برای هر تصویر Q از بردار اول مولفه k ام را در k امین تصویر Q ضرب کرده و با هم جمع میکنیم تا جایی که تصاویر تمام شود حال این تصویر یکی از ۶ تصویر شاخص ما است و به ترتیب برای بردار دوم و ..

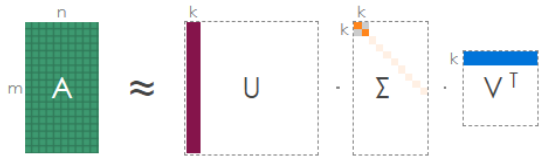
تصاویر حاصل به صورت زیر است که در پوشه ی eigfaces قرار دارند :

چون همه تصاویر دقیقا در مرکز عکس قرار نداشتند شاید برخی از چهره های ویژه شکل های عجیبی به خود گیرد.

۲ فشرده سازی به کمک تجزیه مقادیر منفرد

به این صورت عمل میکنیم که ابتدا تصویر را در ماتریسی به مانند A ذخیره میکنیم سپس به کمک تجریه SVD ماتریس های U ، V و مقادیر ویژه را بدست می آوریم حال میدانیم که مقادیر ویژه نهایتا به تعداد Rank(A) خواهد بود.

اگر ما در ماتریس Σ به تعداد k درایه روی قطر اصلی را نگه داریم از اول و بقیه را صفر کنیم با ضرب U در Σ در ترانهاده ی V مشاهده می شود که حجم کاهش یافته است زیرا ماتریس A به صورت زیر ساخته میشود و با کاهش K صفر های بیشتری در A خواهیم داشت.



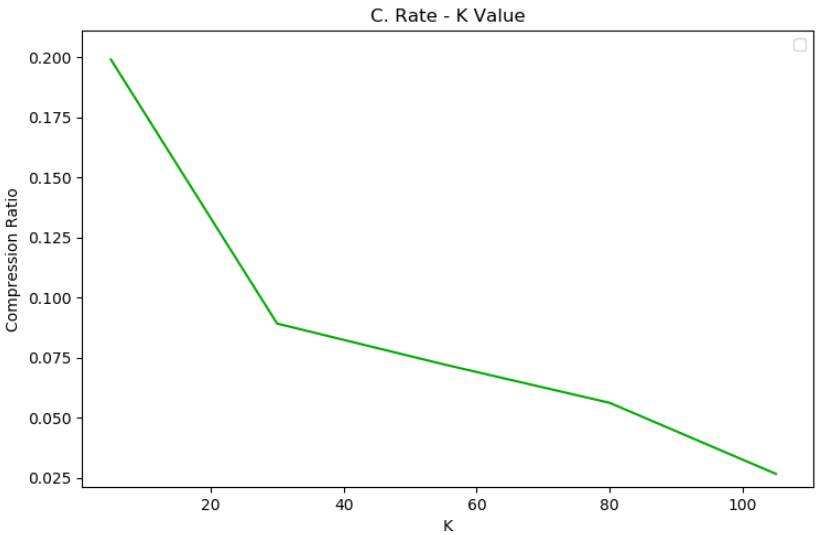
The amount of data needed to store this approximation is proportional to the colored area:

compressed size = $m \times k + k + k \times n = k \times (1 + m + n)$

برای محاسبه حجم میتوانیم ابتدا مجموع درایه های A را حساب کنیم سپس در نهایت دوباره درایه های تصویر نهایی را هم با هم جمع کنیم که تقسیم حجم دومی بر اولی می شود نسبتشان به هم که با کم کردن از یک مقدار حجم کاهش یافته بدست می اید.

گفته های بالا را نمودار زیر بهتر نشان می دهد.

```
Image By k = 5 Created. (Compress Ratio = 19.899% )
Image By k = 30 Created. (Compress Ratio = 8.919% )
Image By k = 55 Created. (Compress Ratio = 7.227% )
Image By k = 80 Created. (Compress Ratio = 5.624% )
Image By k = 105 Created. (Compress Ratio = 2.67% )
```



۳ فشرده سازی به روش ضرایب تبدیل کسینوسی فوریه

به این صورت است که ما برای ذخیره سازی از ضرایب استفاده میکینم و ضرایب را ذخیره می کنیم حالا با تبدیل معکوس میتوانیم خود ماتریس را بازیابی کنیم.

کد محاسبه DCT با توجه به فرمول داده شده در فایل dct.py نوشته شد.

۴ تشخیص چهره به کمک تجریه مقادیر منفرد

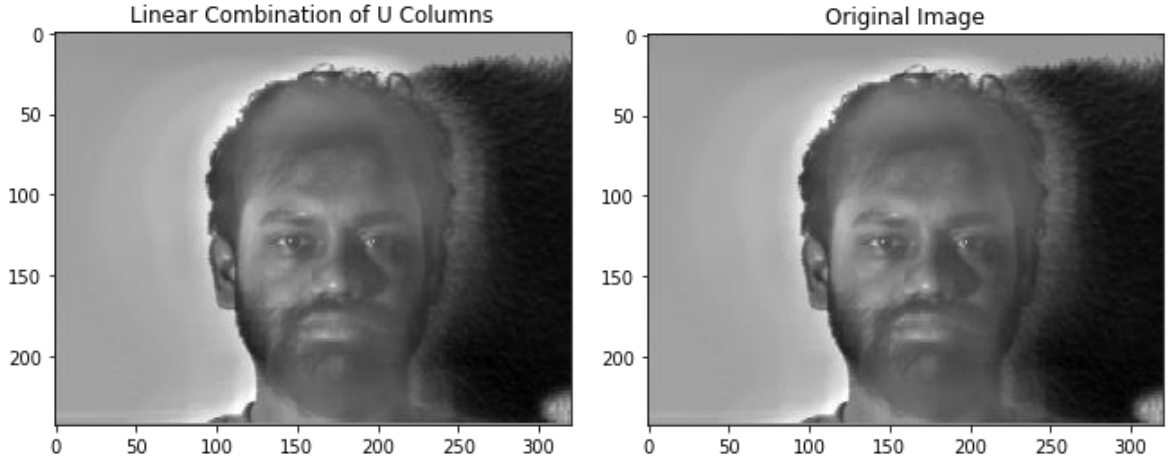
مراحل تا جایی شبیه به بخش اول است فقط بجای تصویر ۲۴۳ در ۳۲۰ با بردار های ۷۷۷۶۰ در ۱ کار می‌کنیم.

توجه کنید مه ماتریس انحراف از میانگین برای داده های train بنده در کد ماتریس A میباشد.

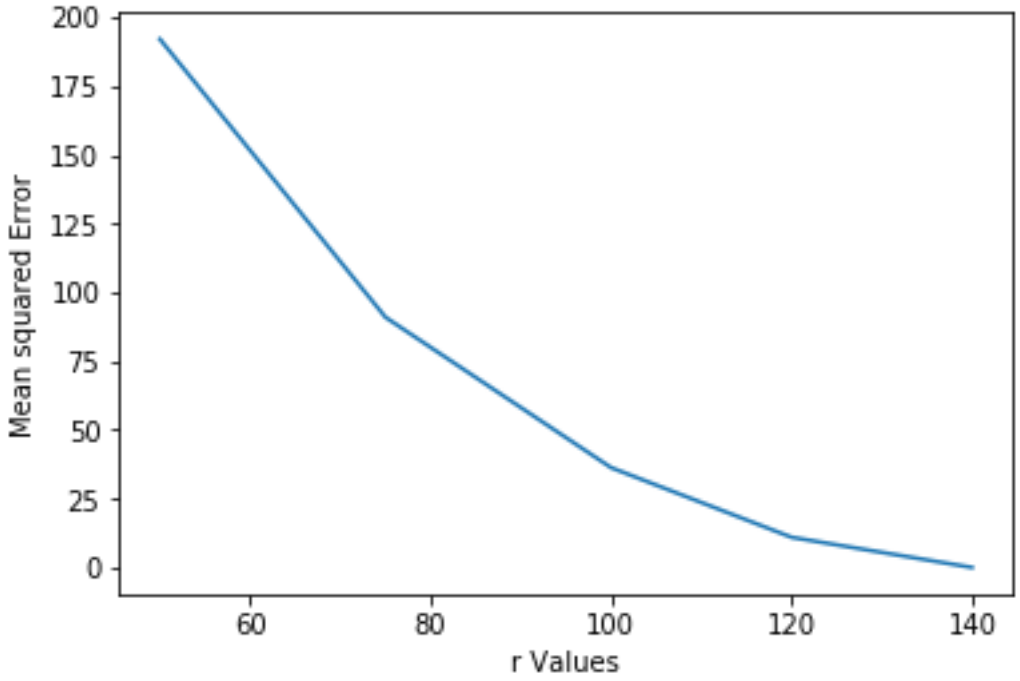
چون تصاویر رو در ستون های A ذخیره مرده ایم و SVD زده ایم پس ستون های U پایه های یکه ای برای فضای برداری چهره های ما هستند ۱۰ ستون اول U به همراه تصویر میانگین داده های Train در فایل svdrec موجود است.

برا اینکه ببینیم هر تصویر با چه ضرایبی ترکیب خطی ستون های U است چون U ارتو نرمال است پس ضرایب برابر ضرب داخلی ستون های U در تصویر مورد نظر است !

در کد یک تصویر به عنوان نمونه بصورت اصلی و یکبار هم به صورت ترکیب خطی از ستون های U با توجه به feature vector آن نمایش داده می شود.



برای بخش بعدی ماتریس A را به صورت شبه SVD به سه ماتریس U (m*r) و S(r*r) و V^T(r*n) تجزیه می‌کنیم و با ضرب اینها ماتریس B را می سازیم. حال با تابع mean_squared_error خطا را پیدا میکنیم و بر حسب r های مختلف رسم میکنیم. که نمودار زیر خروجی آن ست :



تمامی نکات در هر بخش در کامنت های کد نوشته شده است تا ابهامی پیش نیاید.