- درپایتون برای انجام کار ها به پکیج هایی نیاز مندیم. برخلاف متلب با نصب پایتون پکیج ها برای ما نصب نمی شود(همین باعث سبک و راحتی کار با پایتون نیز میشود.به اینصورت که برای هر بار استفاده لازم نیست حجم سنگینی از برنامه را باز کنیم)، ولی با توجه به Open Sourceبودن پایتون به راحتی میتوانیم پکیج های مورد نیازمان را پیدا ،نصب و استفاده کنیم.

```
import numpy as np
import matplotlib.pyplot as plt
import time
from PIL import Image
```

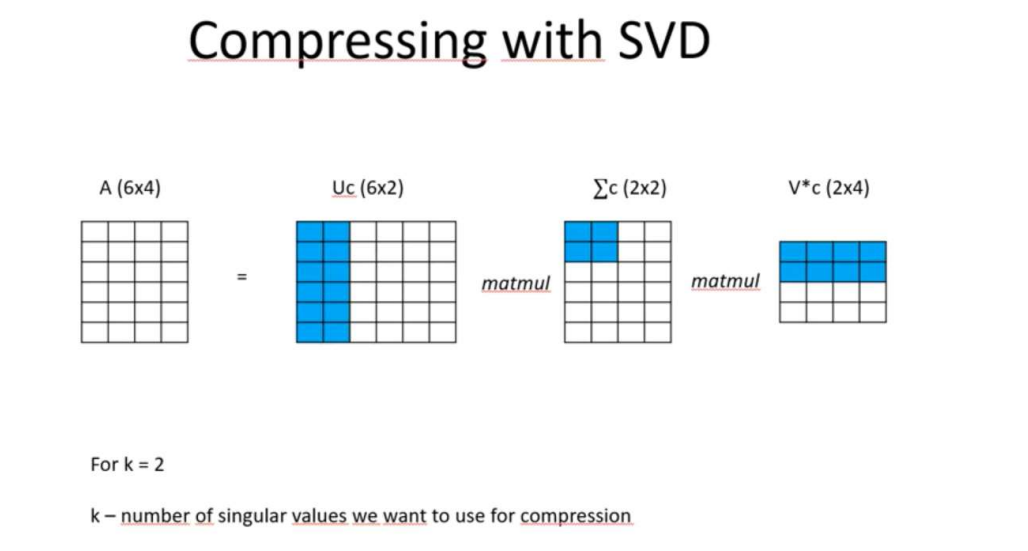
- پکیج اول (numpy)یکی از کتابخانه های ریاضیاتی پایتون میباشد که با داشتن دانش کافی ریاضی میتوانیم کارهای فوق العاده ای با آن انجام دهیم.
- پکیج دوم (matplot) برای رسم نمودار ها و تصاویر از آن میتوانیم استفاده کنیم.
- پکیج سوم و چهارم برای کار های زمانی و مدیریت فایل های سیستم کاربرد دارند

هر عکس در واقع بیانگر یک ماتریس 3 بعدی میباشد که بعد اول و دوم نمابانگر طول و عرض و حاوی پیکسل ها برای نشان دادن عکس میباشد هر پیکسل یک عدد 8بیتی برای مقدار رنگ میباشد.بعد سوم عکس در مود RGBبرای بیان رنگ های به ترتیب قرمز ،سبز و آبی میباشد.

به کد زیر نگاه کنید.ما لایه های بعد سوم عکس را از هم جدا کردیم.

```
# تجزیه ابعاد ماتریس
def openImage(imagePath):
    imOrig = Image.open(imagePath)
    im = numpy.array(imOrig)

    aRed = im[:, :, 0]
    aGreen = im[:, :, 1]
    aBlue = im[:, :, 2]

    return [aRed, aGreen, aBlue, imOrig]
```

در تجریه یک ماتریس به روش مقادیر ویژه 3 بردار به صورت زیر داریم



ماتریس سیگما یک ماتریس قطری حاوی مقادیر ویژه از بیشترین مقدار به کمترین میباشد.برای فشرده سازی همانند عکس بالا میتوانیم عمل کنیم(جدا کردن قسمت آبی از کل ماتریس ها)

```python
# فشرده سازی کانال ها
def compressSingleChannel(channelDataMatrix, singularValuesLimit):
    uChannel, sChannel, vhChannel = numpy.linalg.svd(channelDataMatrix)
    aChannelCompressed = numpy.zeros((channelDataMatrix.shape[0], channelDataMatrix.shape[1]))
    k = singularValuesLimit

    leftSide = numpy.matmul(uChannel[:, 0:k], numpy.diag(sChannel)[0:k, 0:k])
    aChannelCompressedInner = numpy.matmul(leftSide, vhChannel[0:k, :])
    aChannelCompressed = aChannelCompressedInner.astype('uint8')
    return aChannelCompressed
```

در اینجا با دستور svd از ماژول linalg (که مخفف کلمه لاتین جبر خطی میباشد) از کتابخانه numpy مقادیر بردار u,s,v* به دست می آید.

به اندازه حد فشرده سازی از ماتریس ها جدا میکنیم (بقیه را صفر میکنیم ولی اندازه ماتریس تغییر نمیکند)و با دستور های ضرب ماتریسی مقدار جدید را حساب میکنیم.

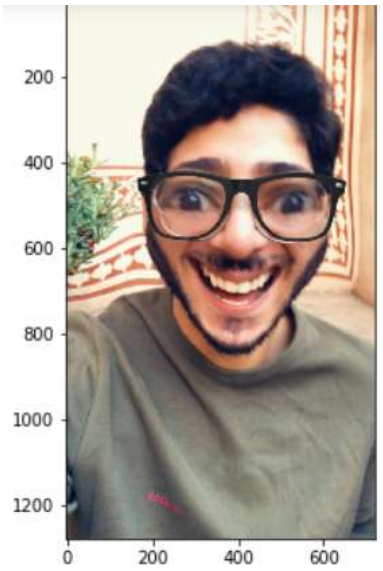
سپس با دستورات مورد نیاز ماتریس های بعد از فشرده ساری را به هم دوباره متصل کرده و یک عکس درست میکنیم.

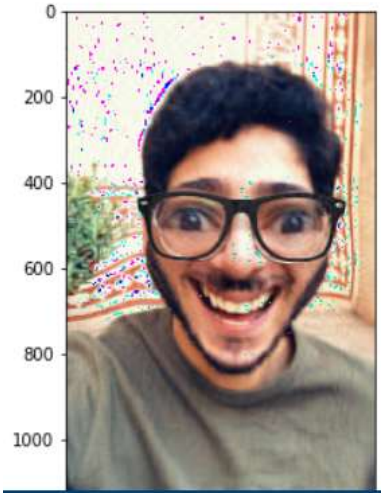
```
*** Image Compression using SVD - a demo

Singular Value Limit :
```

حد مقدار منفرد را تعیین کرده و نتایج را میبینیم.

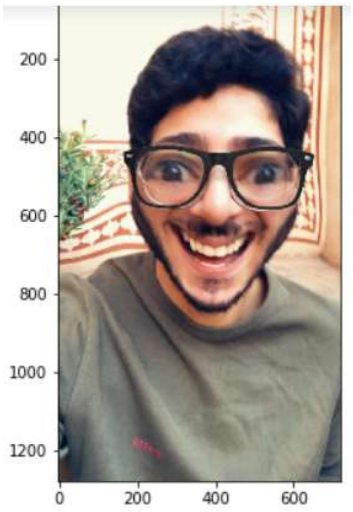
برای 20 داریم:

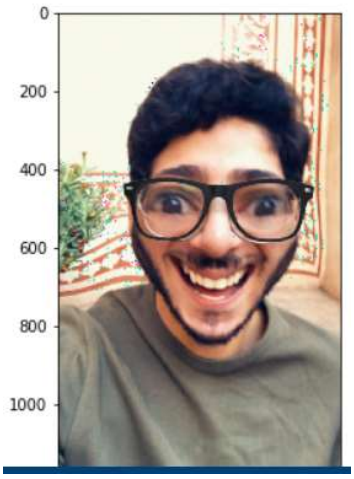
برای 50 داریم:





و برای 100 داریم:





توجه داشته مقدار بیشتر از مقدار منفرد (دراین مورد 179) باعث ارور میشود