

היום נתרגל PCA בעזרת אלגוריתם בשם eigenfaces שמשמש ב- PCA כדי לעשות זיהוי פנים.  
נא להוריד את מאגר התמונות הבא:

<http://cswwww.essex.ac.uk/mv/allfaces/faces94.html>

שימו לב שיש מספר תמונות בכל תיקיה, כדי לעבור על כל הקבצים שנמצאים בתיקייה אפשר להשתמש בפונקציה os.walk שמאפשרת בקלות לעבור על תוכן של תיקייה.

[https://www.tutorialspoint.com/python/os\\_walk.htm](https://www.tutorialspoint.com/python/os_walk.htm)

פונקציה נוספת שניתן להשתמש בה היא: os.listdir

[https://www.tutorialspoint.com/python/os\\_listdir.htm](https://www.tutorialspoint.com/python/os_listdir.htm)

קודם כל נקרא את התמונות בעזרת פונקציה לקריאת תמונה:

יש כמה שיטות - אחת מהן משתמשת בספרייה פשוטה בשם imageio, בפונקציה בשם

```
from imageio import imread  
face = imread('face.png')
```

נרצה להפוך כל תמונה ל-Array בעל מימד אחד בעזרת הפונקציה:

<https://docs.scipy.org/doc/numpy/reference/generated/numpy.reshape.html>

כדי לחבר את כל ה- arrays למטריצה אחת גדולה ניתן להשתמש בפונקציות כמו np.vstack:

<https://docs.scipy.org/doc/numpy-1.15.0/reference/generated/numpy.vstack.html>

עכשיו אפשר לעשות PCA על כל המאגר שהורדנו ולייצג את כל מאגר התמונות על ידי 30 תמונות בלבד.  
נעשה זאת בשני שלבים:

בתחילה נשתמש בפונקציות המוכנות לכך של scikit-learn מהלינקים הבאים:

1. <http://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html>

2. [http://scikit-learn.org/0.18/auto\\_examples/applications/face\\_recognition.html](http://scikit-learn.org/0.18/auto_examples/applications/face_recognition.html)

תוצאות ה-PCA הם eigenfaces של datan, בעזרת reshape המירו אותם חזרה לצורה של תמונה והדפיסו אותם בעזרת imwrite.

נסו לייצג את אחת התמונות בעזרת הבסיס שה-PCA יצר ע"י הטלה של התמונה לבסיס (כפל מטריצה בוקטור) וייצוג התוצאה בעזרת וקטור.

אחרי שהצלחנו זאת - ננסה לממש PCA בעצמנו בעזרת ההוראות מהלינק הבא והפונקציה המתאימה של NumPy:

1. <https://en.wikipedia.org/wiki/Eigenface>

2. <https://docs.scipy.org/doc/numpy/reference/generated/numpy.linalg.eig.html>

תזכרו שצריך לחשב pca על מטריצת covariance

כעת ניתן לקחת תמונה שלנו ולייצג את עצמנו בעזרת eigenfaces.