Principal Component Analysis là gì

Principal Component Analysis là phương pháp dùng để giảm số chiều của dữ liệu bằng cách tìm các chiều là thành phần chính của dữ liệu, từ đó ta có thể biểu diễn dữ liệu theo công thức:

$$X = X_0 + z_1 * X_1 + z_2 * X_2 + ... + z_k * X_k \quad (1)$$

Với

1. k là số chiều của dữ diệu
2. X_0 là means của training data
3. X_i là basic function
4. z_i là toạ độ của điểm dữ liệu

Dữ liệu qua thuật toán PCA trở thành các điểm toạ độ có thể sử dụng để so sánh hay lưu trữ...

Encode

Khi áp dụng thuật toán ta có U là ma trận lưu basic functions, từ đó có thể tích toạ độ z của điểm dữ liệu bằng cách chiếu điểm dữ liệu lên các trục số $z = U^T * x$ với $x = X - X_0$

Decode

Với z là toạ độ của điểm dữ liệu, khi decode dữ liệu ta chỉ cần thay toạ độ của dữ liệu vào công thức (1)

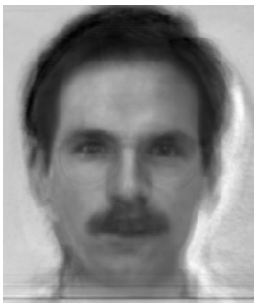
Thực nghiệm trên code

Áp dụng với bài toán encode và decode T ảnh Với mỗi K thuộc [1,10,20,50,100], ta có ảnh decode như sau:

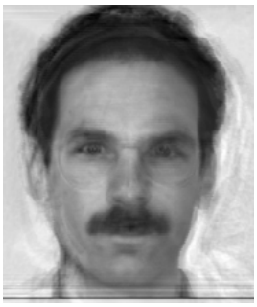
origin picture

K = 1, loss = 33404.46579972785

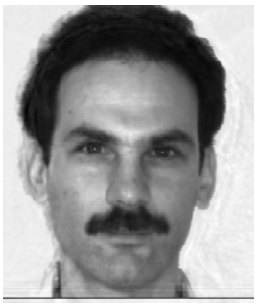
K = 10, loss = 31214.950574420687



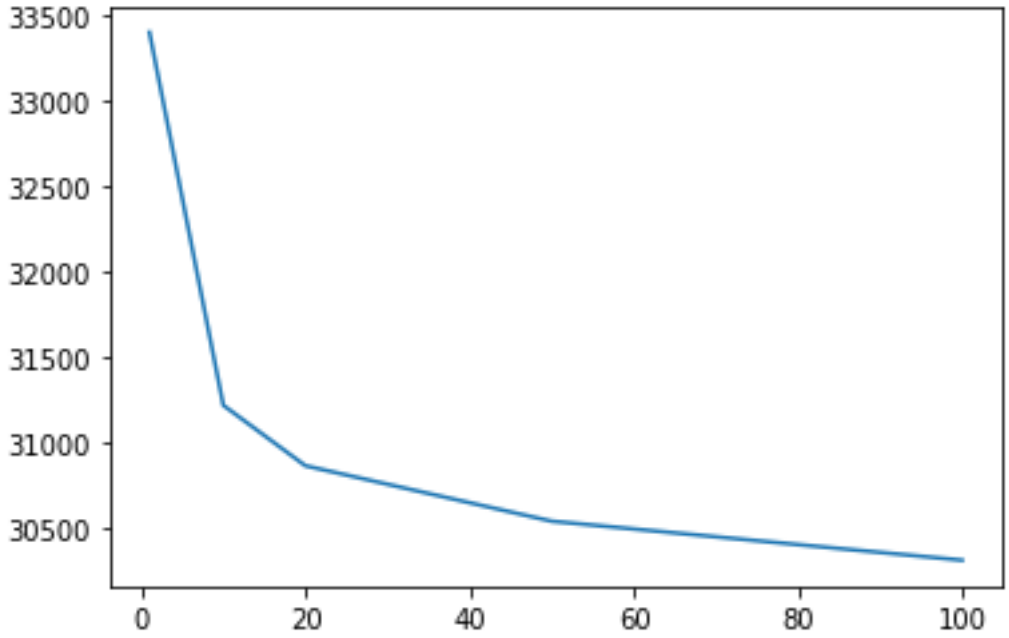
K = 20, loss = 30860.068222190162



K = 50, loss = 30534.718245444663



K = 100, loss = 30305.81835208503



reconstruction errors

Quiz

1. coordinate vector $z = U'.dot(x)$ với $x = im - mean$
   Ta có: $x = z_1 * X_1 + z_2 * X_2 + ... + z_k * X_k$
   Vì các basic vector vuông góc với nhau nên khi nhân 2 ma trận ta sẽ chiếu được giá trị của điểm lên các trục

2. tính compression ratio cho ví dụ eigenface hôm nay nếu chỉ truyền 1 coordinate vector length-100 cho mỗi hình
   dung lượng phải lưu cho T hình gốc là ma trận H*W*T (trong đó H và W là kích thước 1 hình
   dung lượng phải lưu cho T hình đã nén là K ma trận basis vector có kích thước H*W*K và K vector coordinate có kích thước K*T
   vậy tỉ lệ nén với K = 100 khi nén T hình là

$$\frac{H * W * 100 + T * 100}{H * W * T}$$