



# 수치해석 HW#14

K-Means Clustering & Mean Shift Clustering

Python(opencv) 사용

컴퓨터소프트웨어학부 2015005187 최철훈



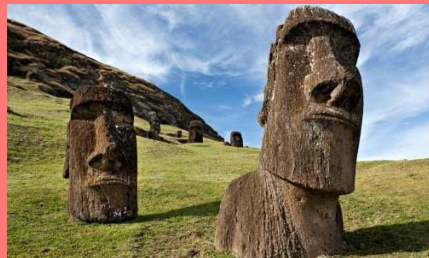
# 목차

1. Lab포맷으로 변환
2. Mean Shift Clustering
3. K-Means Clustering
4. 비교
5. 마무리

# Lab포맷으로 변환 변환 & 크기조절

3개의 이미지를 선택하여 Lab포맷으로 변환하였다. Mean Shift Clustering을 수행할 때 이미지의 크기가 크면 너무 시간이 오래걸리므로 200X100정도의 사이즈로 줄였다.

원본 이미지



L



a



b



# Mean Shift Clustering $h$ 정하기

각각의 이미지를 3차원 공간좌표에 scatter해보고 클러스터의 개수가 15~20 사이가 되도록  $h$ 값을 조절하였다.

$h$ 값 : Image1 – 14

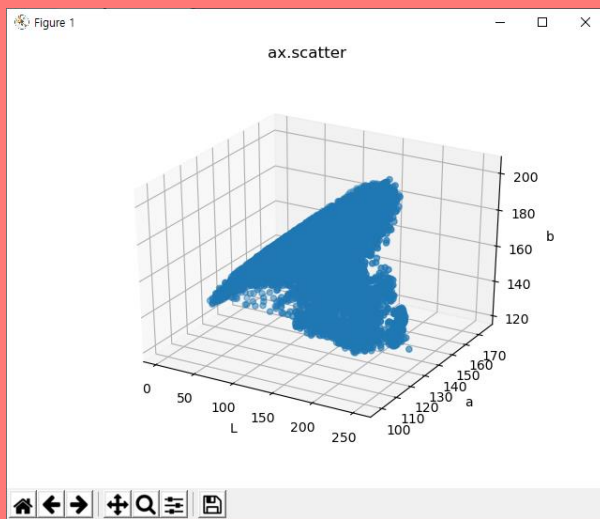


Image2 – 9

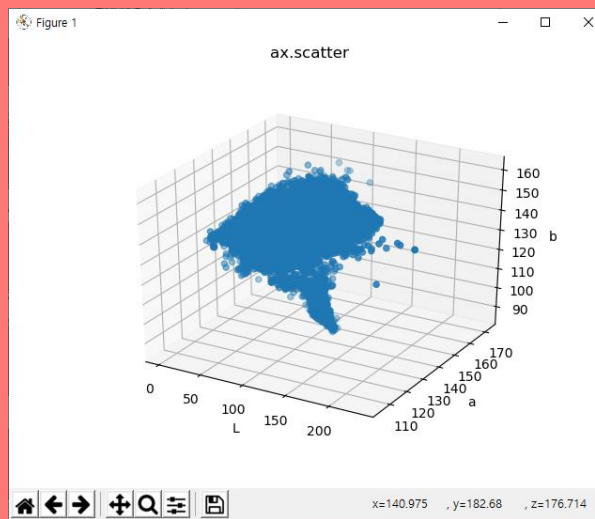
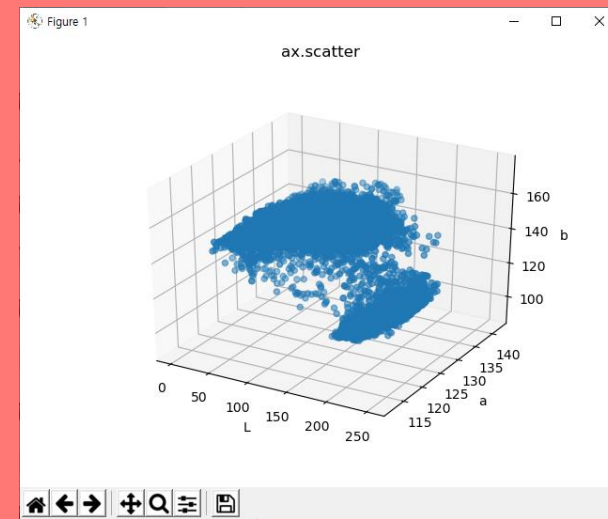


Image3 – 11



# Mean Shift Clustering 과정 및 결과

앞서 구한  $h$ 값으로 Mean Shift Clustering을 각각의 이미지에 대하여 실시하였고, 다시 RGB포맷으로 변환하여 시각화하였다. 아래는 Mean Shift Clustering을 실시한 후의 이미지 결과이다.

클러스터 개수

Image1 – 17



Image2 – 17



Image3 - 19



# K-Means Clustering 과정 및 결과

앞선 Mean Shift Clustering으로 나온 클러스터의 개수를 K로 두고 K-Means Clustering을 진행하였다. 그 후 RGB포맷으로 변환하여 시각화하였다. 아래는 K-Means Clustering을 실시한 후의 이미지 결과이다.

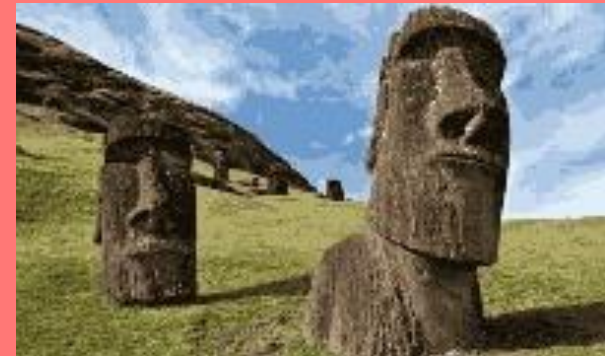
K = 17



K = 17



K = 19





# **비교** Mean Shift vs K-Means

원래 이미지와 Mean Shift, K-Means Clustering을 진행한 이미지들을 비교해보았다. Mean Shift보다는 K-Means Clustering이 원본 이미지에 더 가까운 것을 알 수 있다.

**원본 이미지**



**K-Means**



**Mean-Shift**



# 마무리 느낀점

1. Mean Shift Clustering에서 픽셀의 위치값은 제외하고 오로지 L, a, b성분만을 Clustering하였기 때문에 K-Means Clustering만큼 잘 안나왔다고 생각한다. 실제로 픽셀의 위치값까지 Clustering을 하였을 때 원본 이미지에 가깝게 나왔다.
2. Mean Shift Clustering에서 h값을 크게하면 클러스터링의 개수가 적어졌고 h값을 작게하면 클러스터링의 개수가 많아졌다.
3. K-Means Clustering에서는 K값을 크게 할수록 더 원본과 가까운 이미지를 얻을 수 있었다.



# 마무리

감사합니다.

[https://github.com/cheol-hoon/Numerical\\_Analysis](https://github.com/cheol-hoon/Numerical_Analysis)