



# HW14

2016024766

김서현



# Input Images

---



# Mean shift clustering 구현

---

- 우선 이미지를 불러와서 Lab-1976 color space로 변환시켰습니다.
- Mean shift clustering을 하기 위해서 sklearn의 MeanShift() 함수를 사용했습니다.
- Bandwidth를 바꿔가면서 여러 번 Mean shift clustering을 해서 Bandwidth를 적절한 값으로 조정했습니다.
- Mean shift clustering을 마치고 나온 cluster의 개수를 따로 저장하였습니다.
  - *K Means clustering의 k로 사용하기 위해서*
- Mean shift clustering을 통해 만들어진 segmented image를 다시 RGB로 변환하여 출력해보았습니다.

# K Means clustering 구현

---

- 우선 이미지를 불러와서 Lab-1976 color space로 변환시켰습니다.
- K Means clustering을 하기 위해서 sklearn의 KMeans() 함수를 사용했습니다.
- Mean shift clustering을 해서 생긴 cluster의 개수를 K로 설정했습니다.
- K Means clustering을 통해 만들어진 segmented image를 다시 RGB로 변환하여 출력해보았습니다

# Clustering 결과 #1

original



Mean shift



K means



Bandwidth: 20  
# of Clusters: 7

original



Mean shift



K means



Bandwidth: 15  
# of Clusters: 12



# Clustering 결과 #2

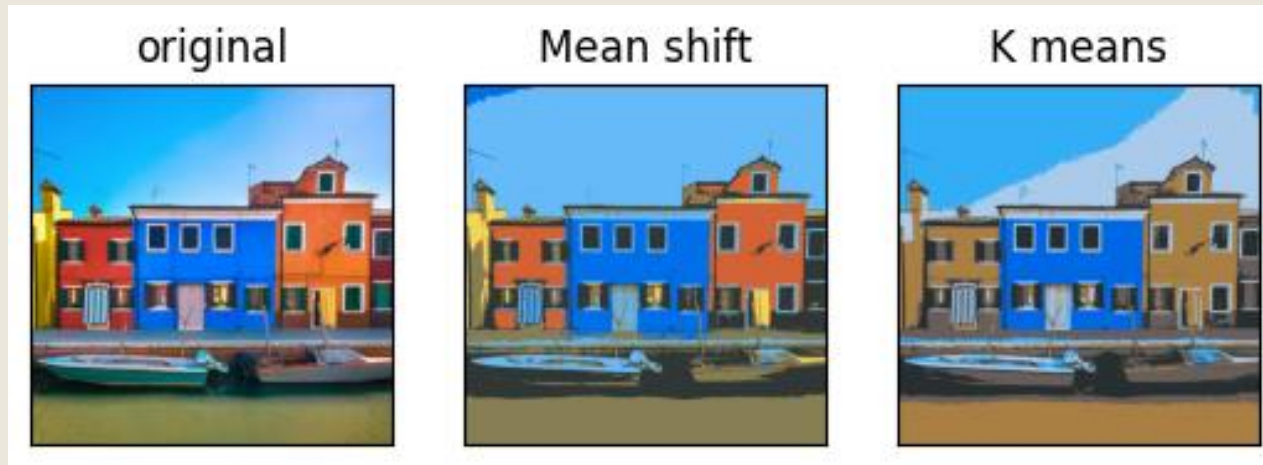


Bandwidth: 22  
# of Clusters: 5



Bandwidth: 15  
# of Clusters: 14

# Clustering 결과 #3



Bandwidth: 40  
# of Clusters: 6



Bandwidth: 15  
# of Clusters: 20

# 결과 분석

---

- 결과#1과 #3을 봤을 때, bandwidth가 크고 cluster의 개수가 적을 때에는 Mean shift clustering의 결과가 K Means clustering의 결과보다 더 원본에 가깝다는 것을 알 수 있습니다.
- Cluster의 개수가 일정 개수(대략 10개) 이상일 때에는 두 clustering 방법의 결과가 모두 원본과 비슷하게 나오는 것을 알 수 있습니다.
- Cluster의 개수가 같음에도 불구하고 두 clustering 방법의 결과가 조금씩 다르게 나오는 것을 보아, Input image들의 분포는 가우시안 분포가 아니라는 것을 알 수 있습니다.