



Palette Studio
- *Color Visualization*



01
Color Space

Color Space

RGB



Red / Green / Blue

빛의 삼원색

TV와 컴퓨터에서 활용

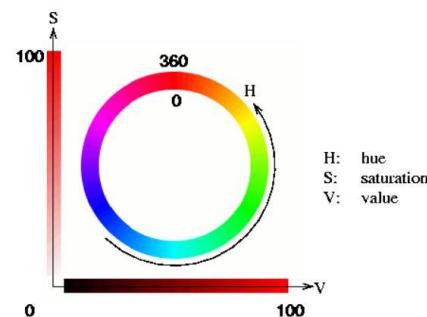
사람이 인식하는

거의 모든 색상 표현 가능

모든 필터에 적용 가능

직관적이지 않음

HSV

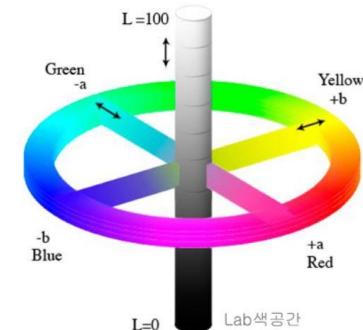


Hue / Saturation / Value

색상, 채도, 명도

색을 가장 직관적으로 표현

Lab



Lightness / a* / b*

L - 밝기

a* (-) 초록 — 빨강 (+)

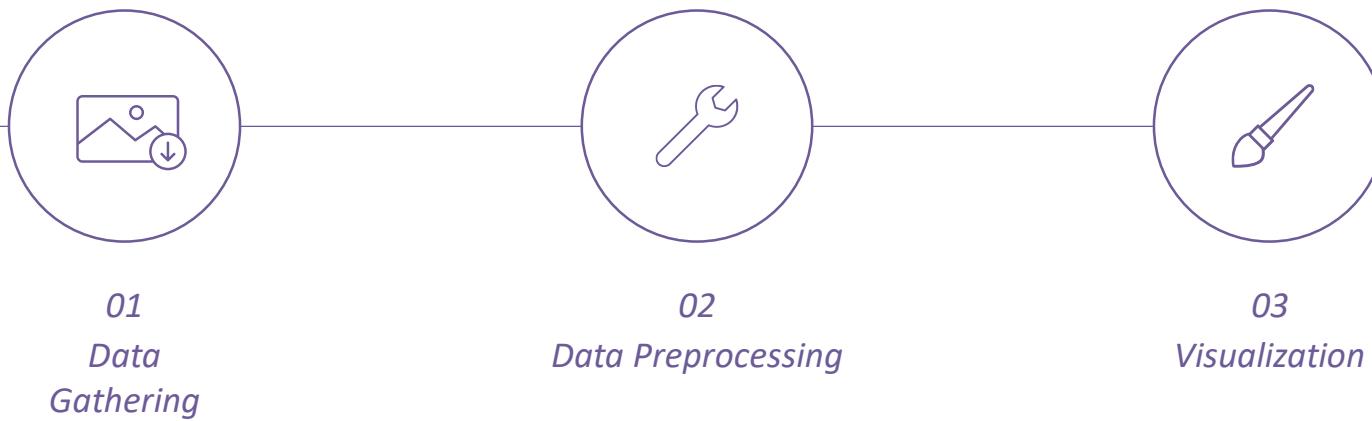
b* (-) 파랑 — 노랑 (+)

인간의 색 지각에 대한 연구를
바탕으로 이루어진 색 공간
모니터나 프린터에 좌우되지
않는 독립적인 방법으로 색상
구현



02
Emoticon

Emoticon



Emoticon

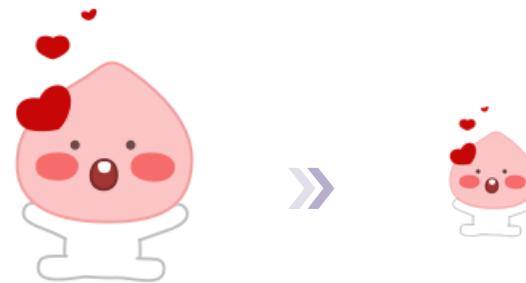
01 - Data Gathering

python의 HTTP module [request](#) 및 URL package [urllib](#) 사용



카카오 이모티콘샵에서 최신순으로 모든 이모티콘을 수집
총 2,384 종류의 61,889개의 이모티콘 데이터 수집

이모티콘샵 상에서 gif 데이터 수집 불가 → 각 이모티콘을 [png](#) 형태로 다운로드



python image library (PIL) 사용 → [이미지 사이즈 통일](#)하여 연산량 감소

Emoticon

02 - Data Preprocessing

scipy.misc의 imread method 사용



image



```
array([[[255, 212, 212, 255],  
       [255, 212, 212, 255],  
       [255, 206, 206, 255],  
       ...], dtype=uint8)
```

numpy array

(255, 157, 157, 0.6)

RGBA



(255, 196, 196)

RGB



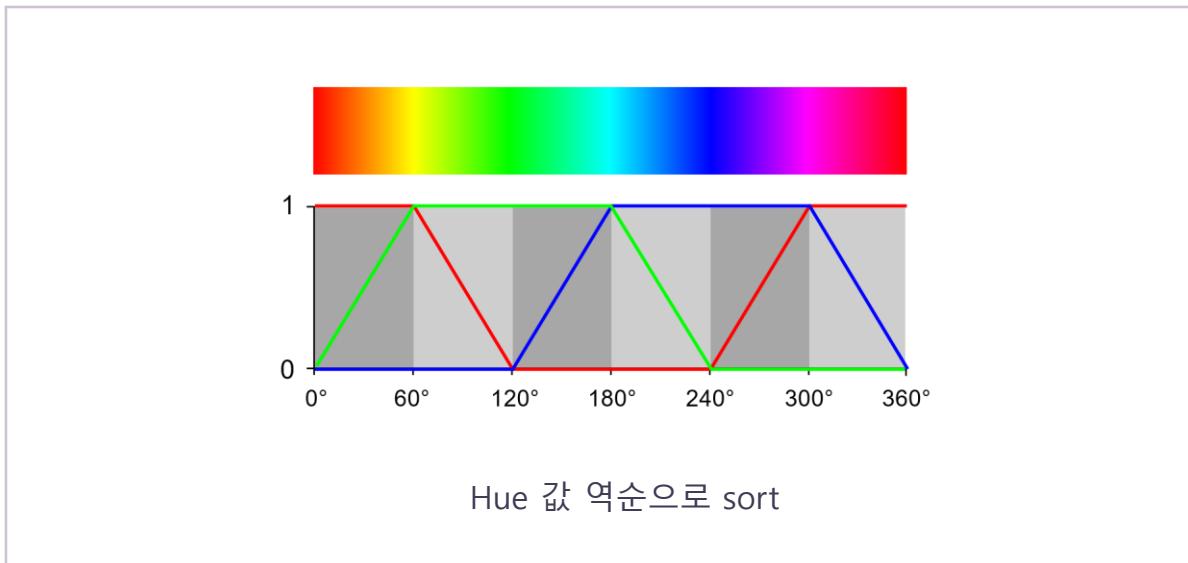
(0, 23.1, 100)

HSV

색공간 module colormath 사용하여 색공간 변환

Emoticon

02 - Data Preprocessing



시각화하기 위하여 다시 RGB 색 공간으로 변환

Emoticon

03 - Visualization





(심한 말)



와아아아아아아



고맙습니다



미안...



아하



굉장히 예민



흉악을 물려라

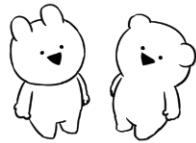


가자! 애들아

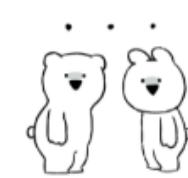


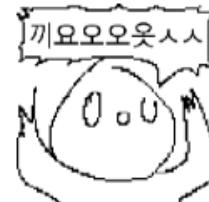
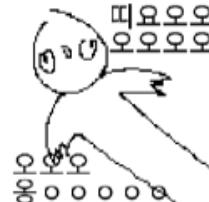
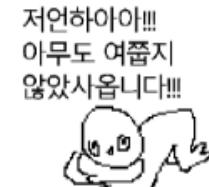
CCCCCCCC





싫어 안해 못해

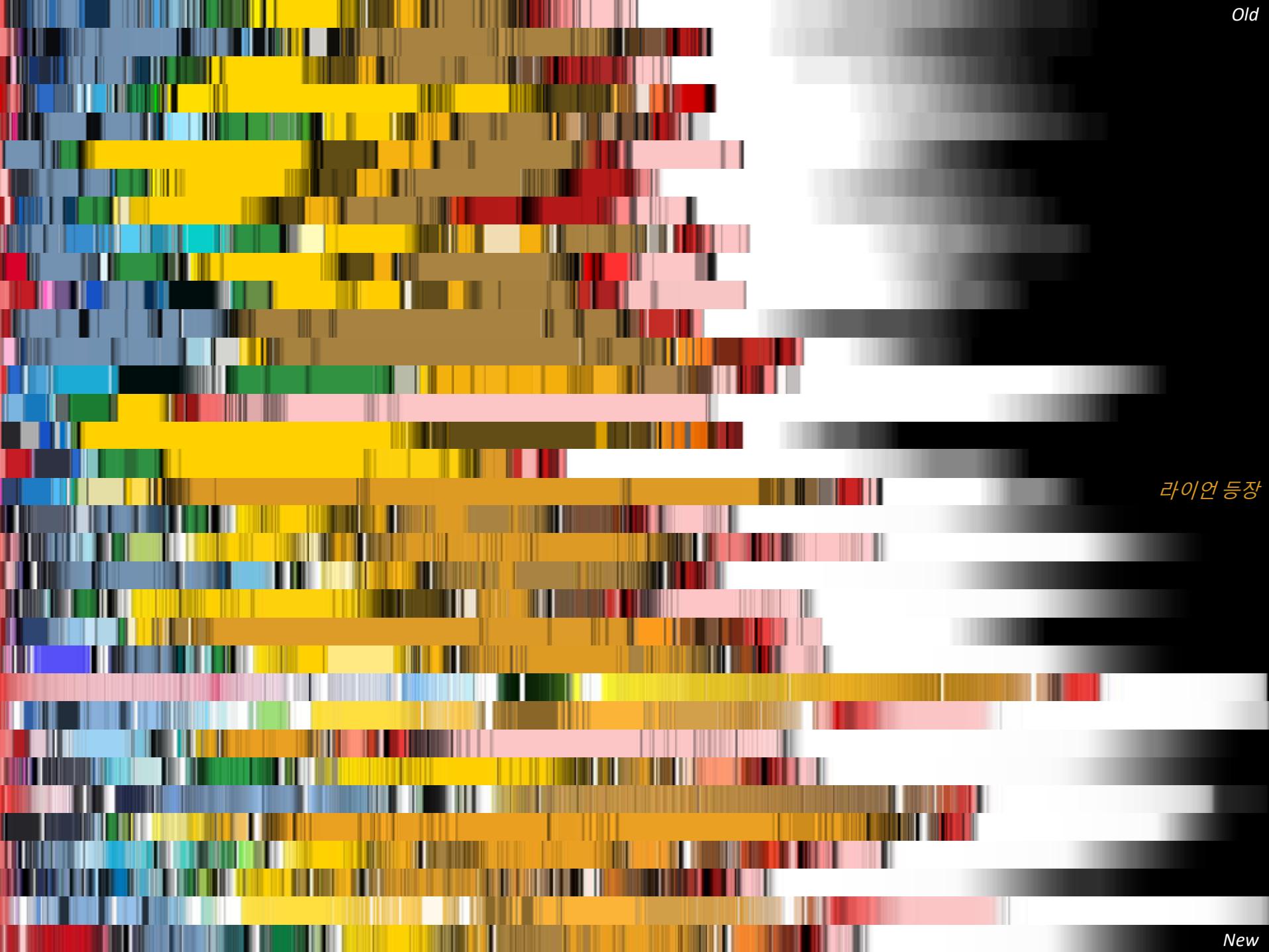






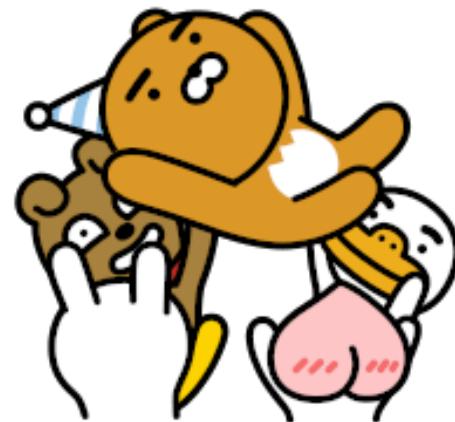


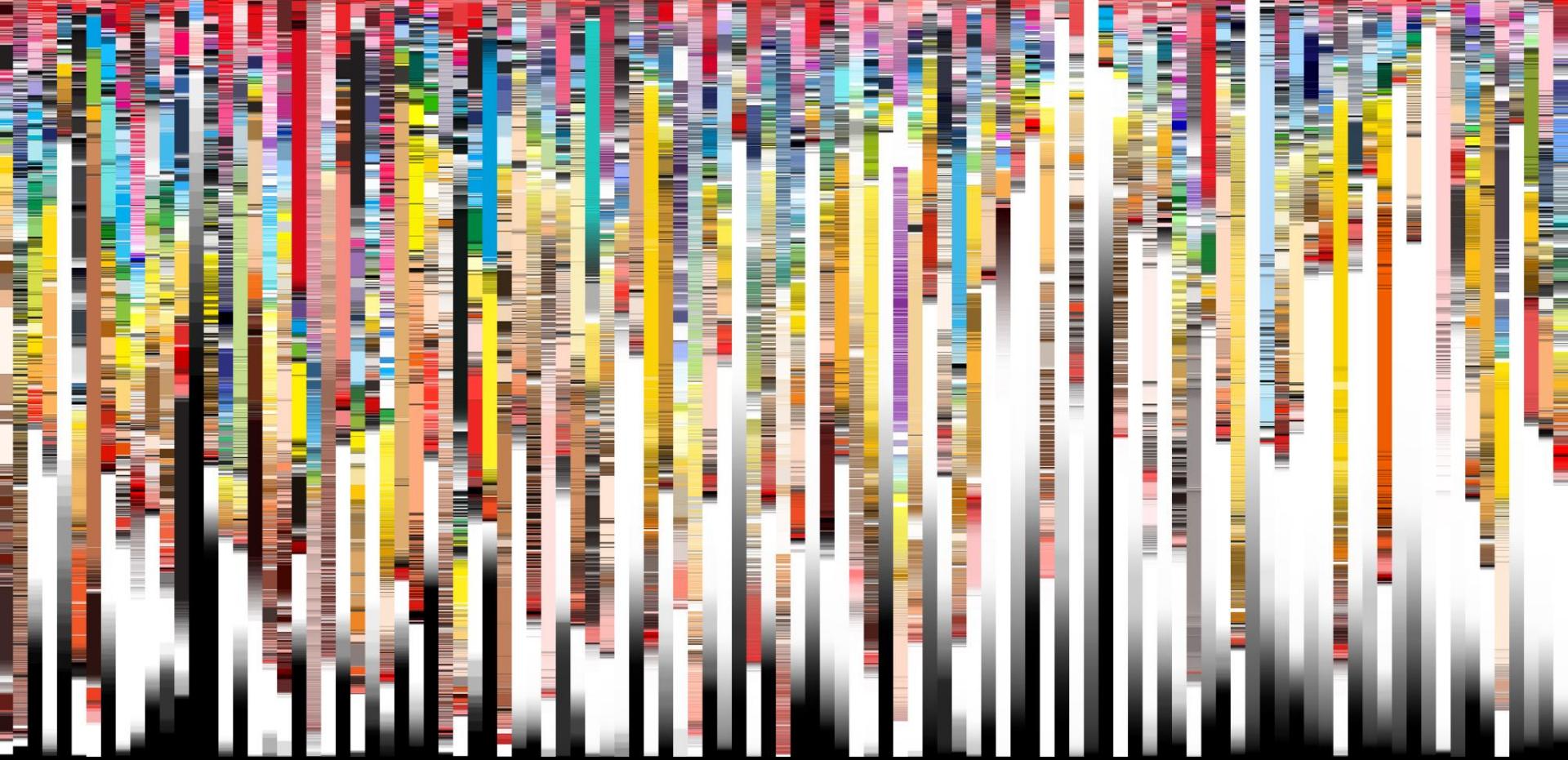
Old



New

라이언의 독주와 갈 곶 잃은 프렌즈들..





Old

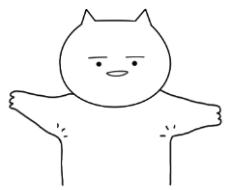
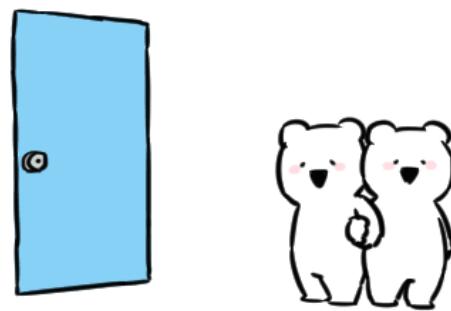


New



오버액션토끼
등장!

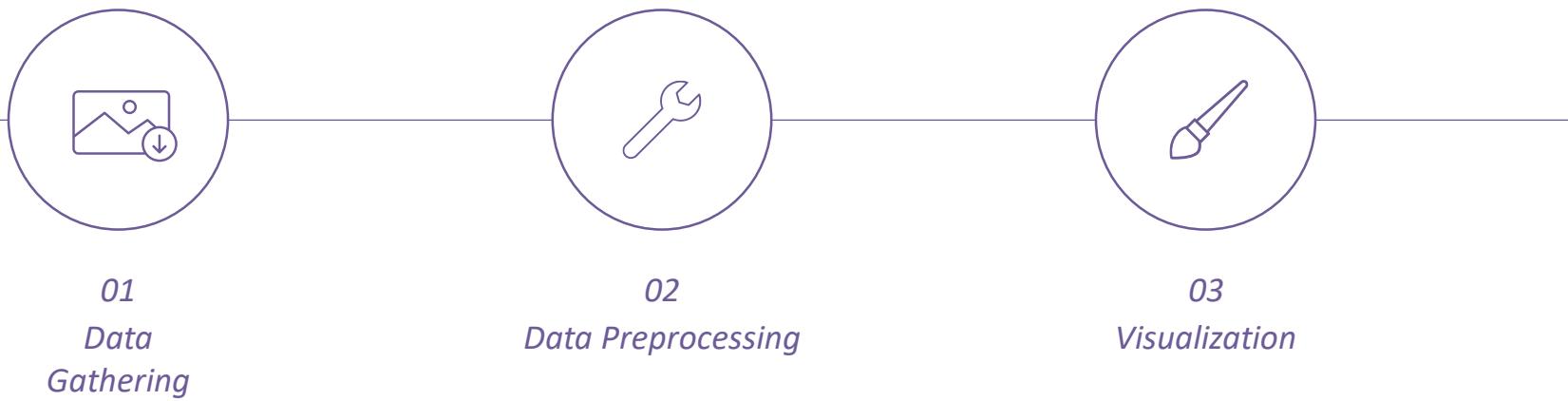
이후 수많은 흰둥이들의 등장!





03
도시 대표 색상

도시 대표 색상



도시 대표 색상

01 - Data Gathering

한 장의 사진이 도시를 대표할 수 없음

여러 장의 사진?



도시 색상과는 무관한 사진이 수집될 가능성 존재

관광지를 소개하는 영상으로 대체!



OpenCV 와 K-Means Clustering 사용



N개 간격으로 프레임
추출



매 프레임마다 대표색상 추출



도시 대표 색상

02 - Data Preprocessing



매 프레임마다 추출된 대표색상으로 Color Barcode 생성

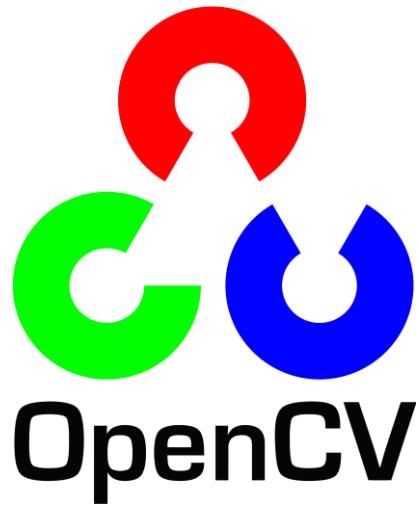


위 바코드에서 다시 K-Means로 12개의 대표색상 추출!

도시 대표 색상

02 - Data Preprocessing





OpenCV

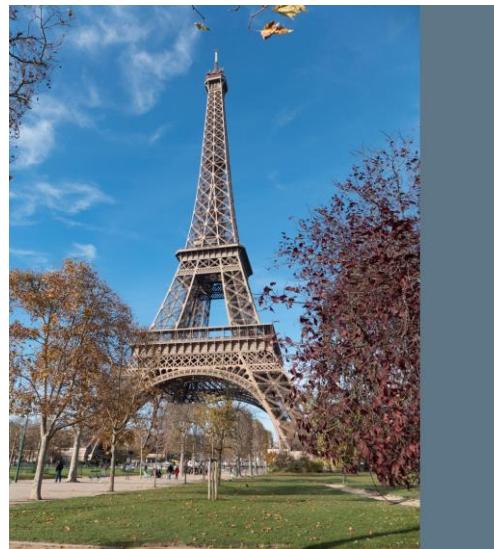
- Intel에서 개발한 영상처리 라이브러리
- 오픈소스로 개발
- 빠르고 안정적인 CV 알고리즘 집합체

도시 대표 색상

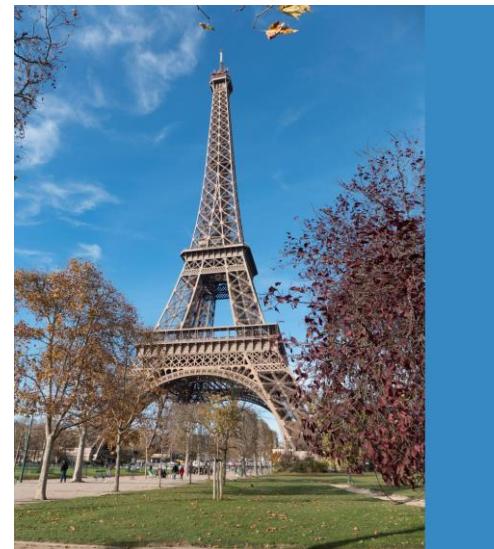
02 - Data Preprocessing

이미지의 대표 색상?

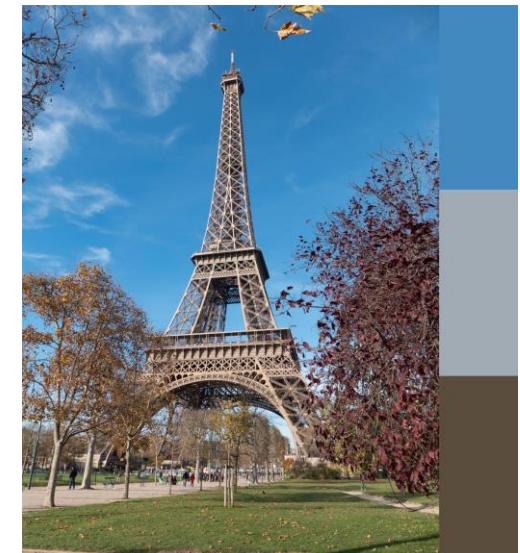
RGB
평균값?

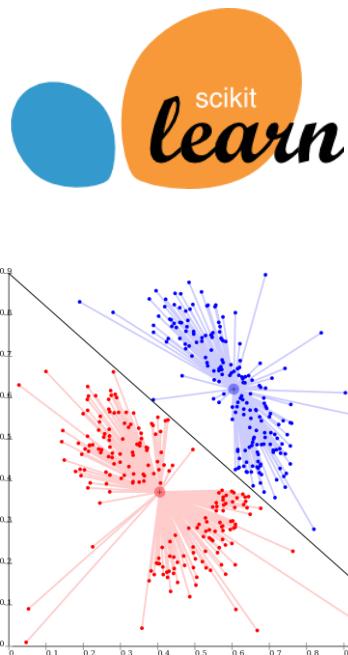


가장 빈도수가 높은 색?



K-Means!

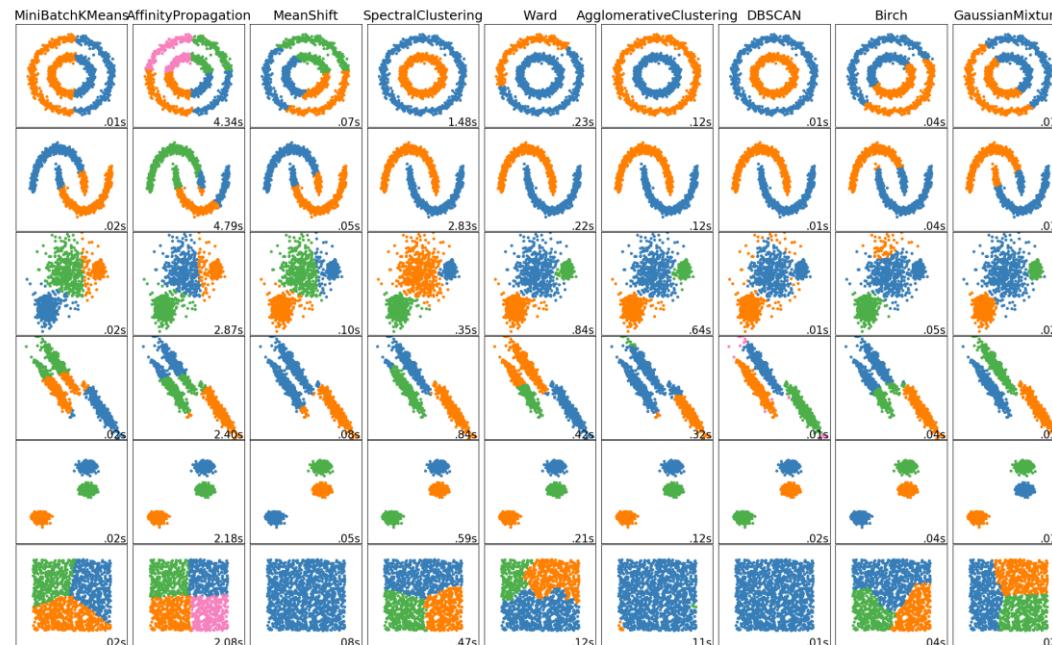




K-Means Clustering

- 가장 대표적인 클러스터링 기법
- K개의 중심점 기준!
- 중심점에서 각 점간 거리 합이 최소가 되는 중심점의 위치 찾기
- 중심점 = 각 군집 대표 평균(Mean)

다른 클러스터링 기법?



적절한 클러스터링 기법 선정은 데이터 모양에 달림...

색상데이터는 원형 군집형태, K-Means로 충분히 유의미한 결과 도출 가능!

K-Means의 문제점?

너무 느리다....!

답은 “Mini-Batch” 다!

SGD(경사하강법) + Batch(가중치 갱신) = Mini Batch

수 백 개의 프레임에 적용할 수 있을 정도의 속도!

03 - Visualization

남아메리카

South America

Peru



Amazon



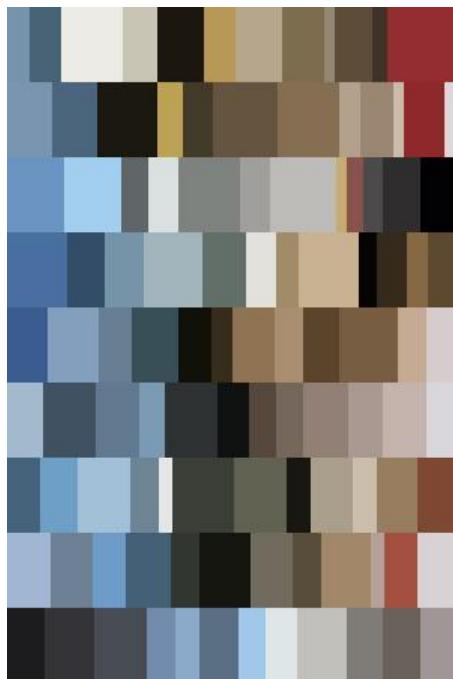
Brazil



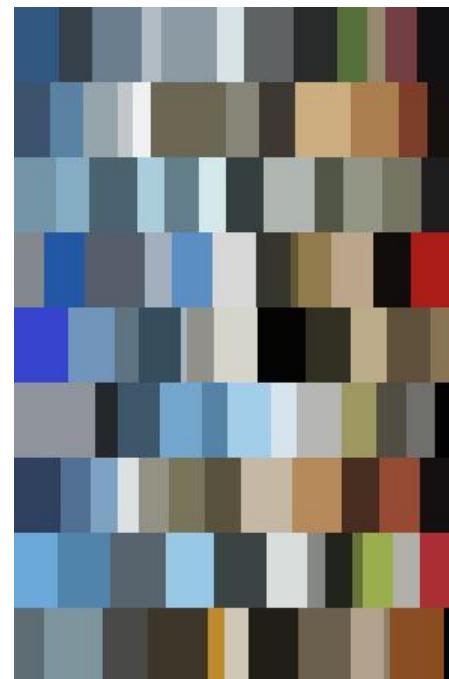
오세아니아

Oceania

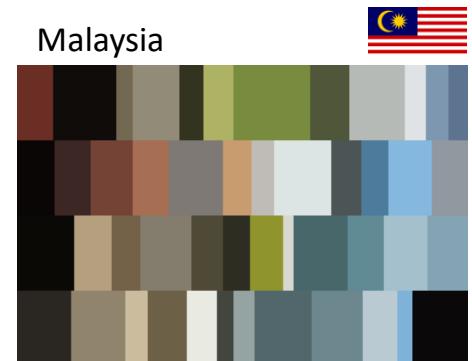
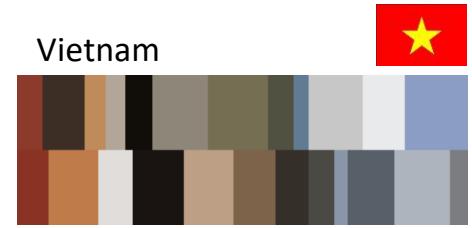
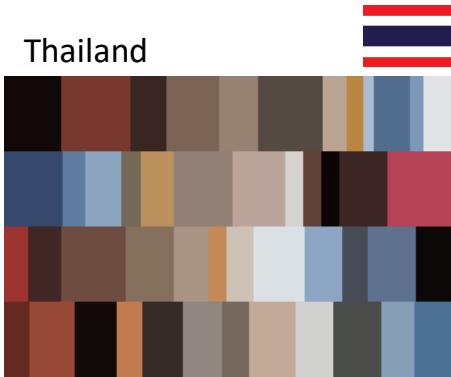
Australia

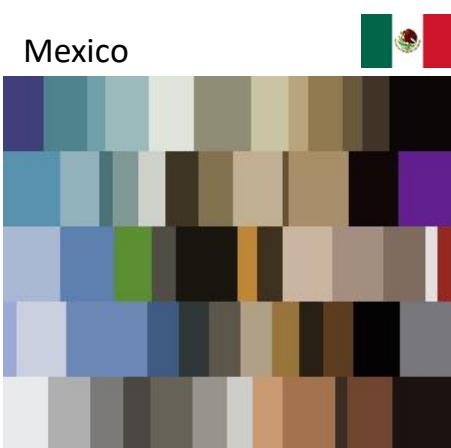
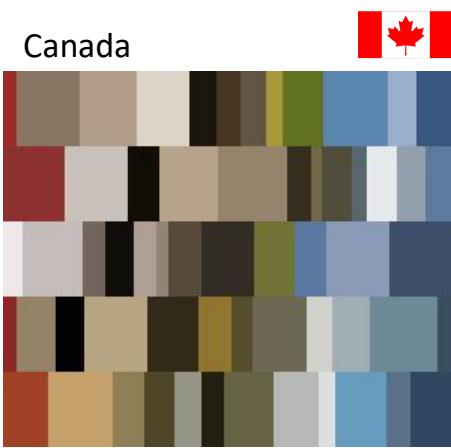


New Zealand



동남아시아 Southeast Asia

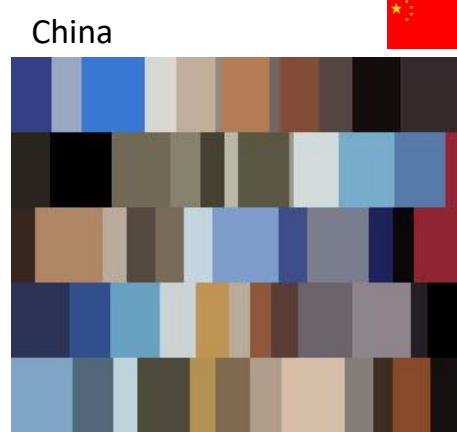




북아메리카
North America

동북아시아

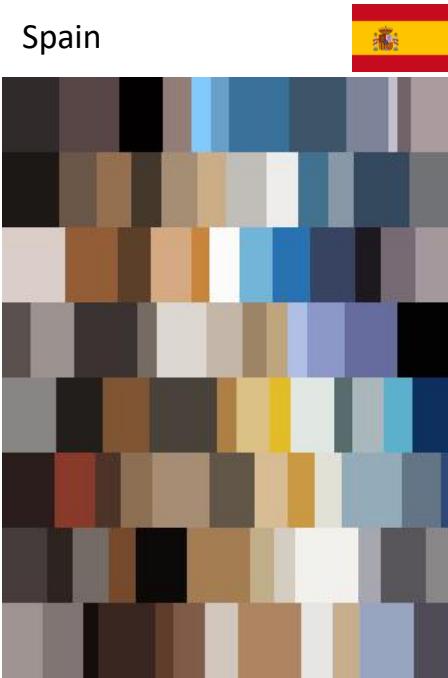
Northeast Asia



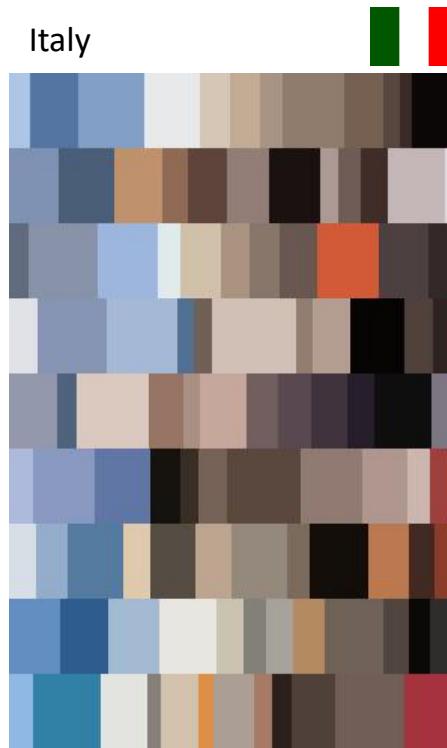
유럽

Europe

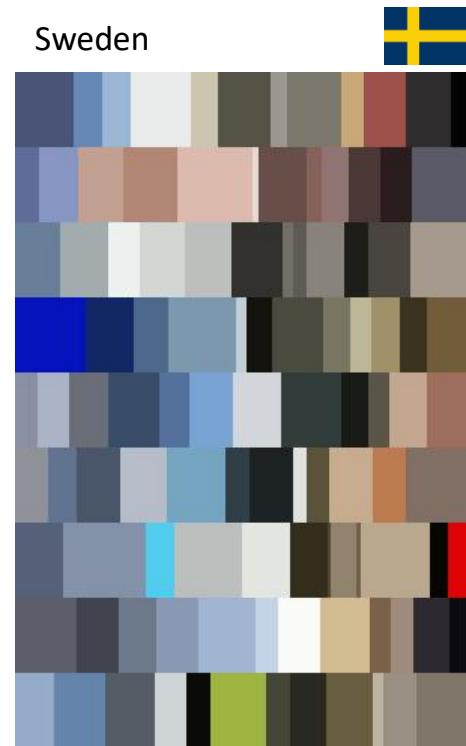
Spain



Italy



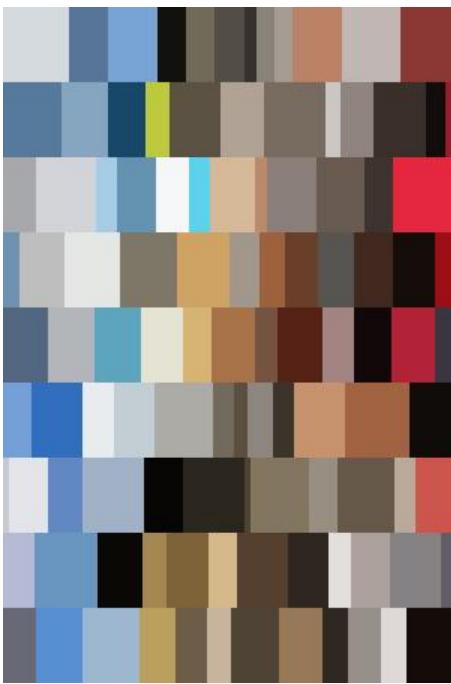
Sweden



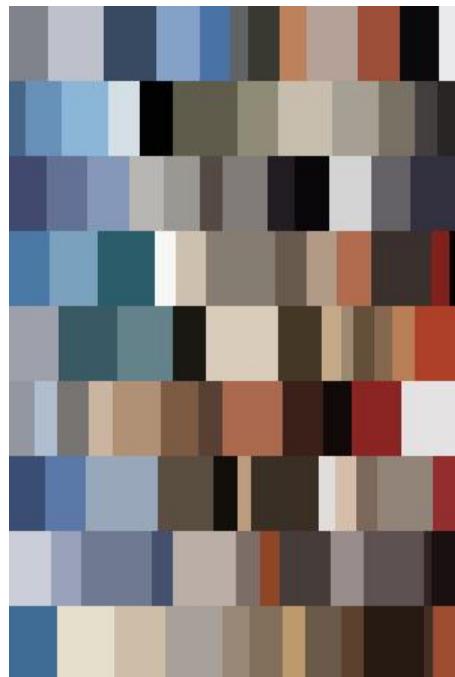
유럽

Europe

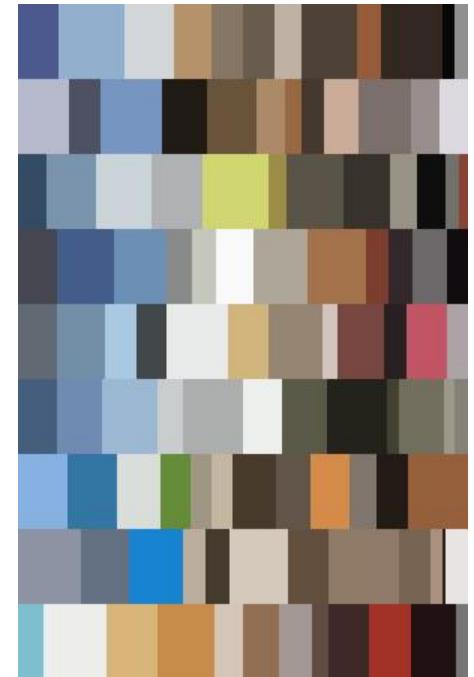
England

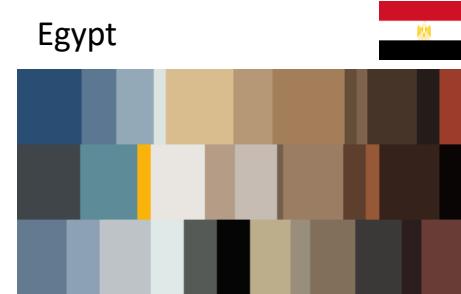
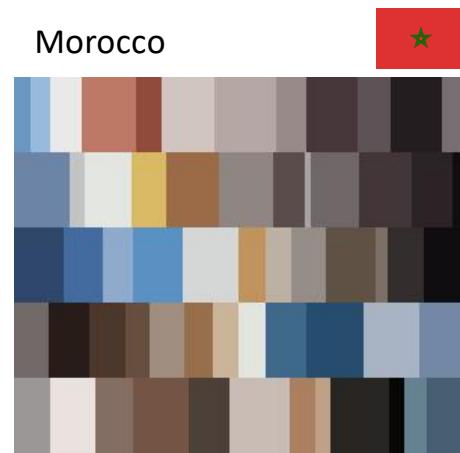
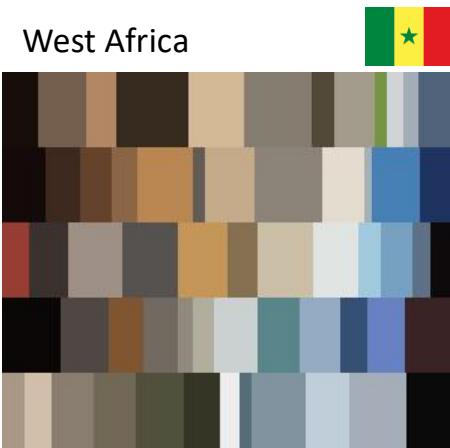


France



Germany



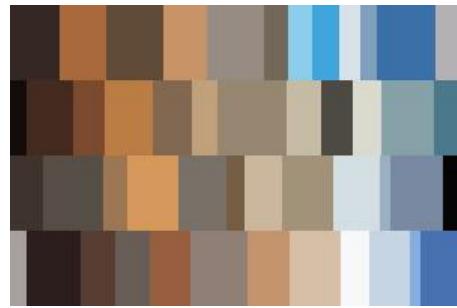


아프리카
Africa

아프리카

Africa

Namibia



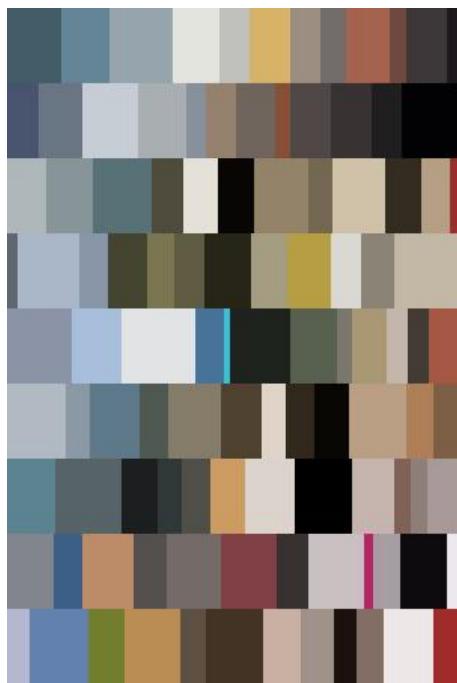
Central Africa



South Africa



South Korea





04
Img2Palette

Image to Palette



Image to Palette

Original



After Clustering ($K = 6$)

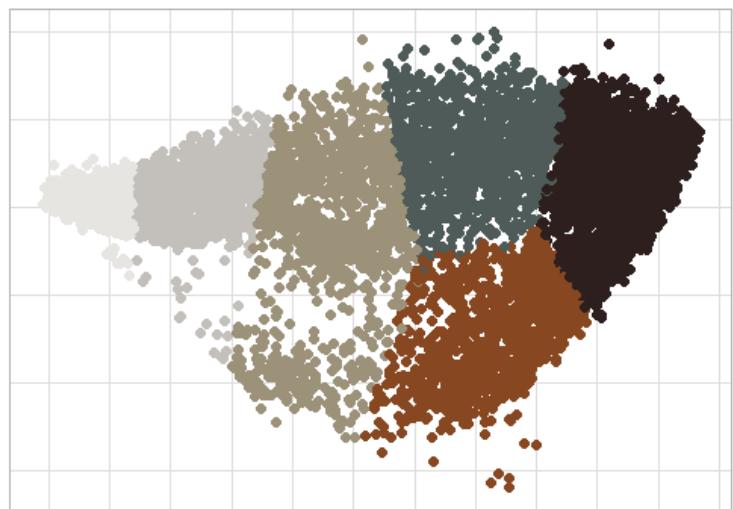
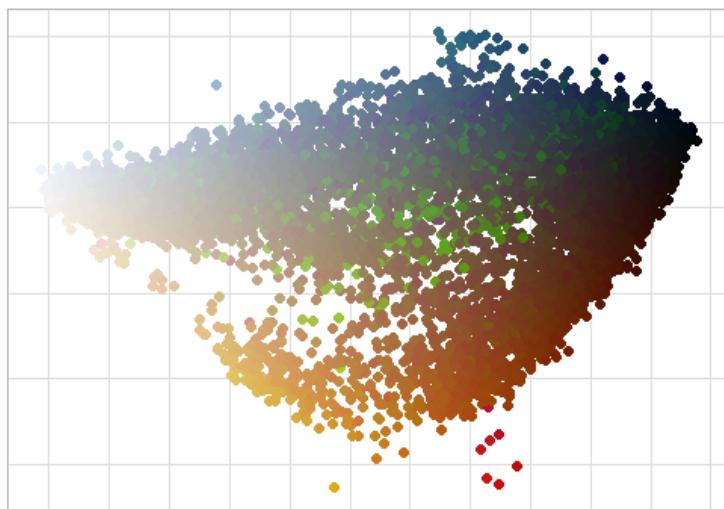


Image to Palette

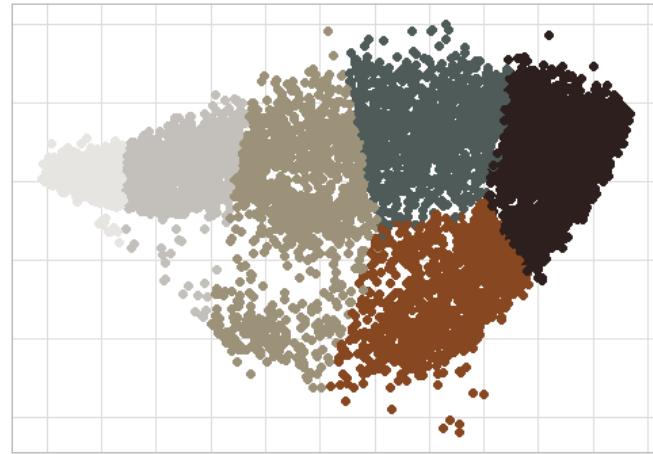
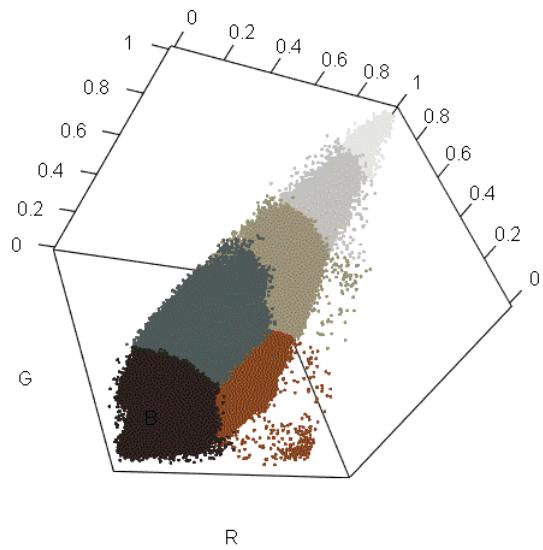
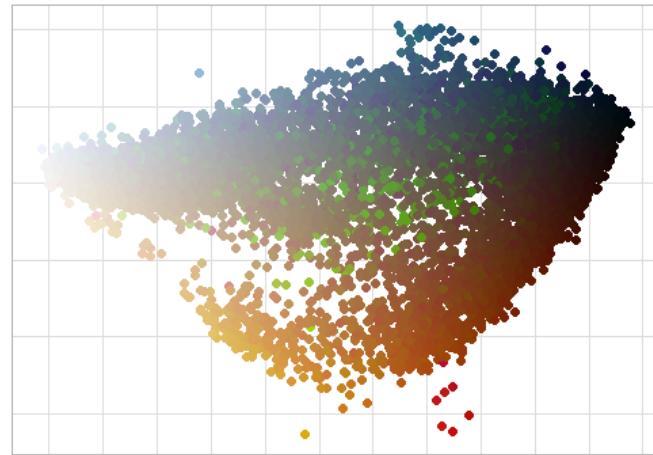
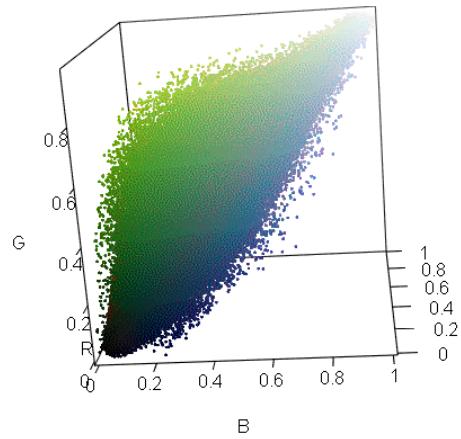


Image to Palette

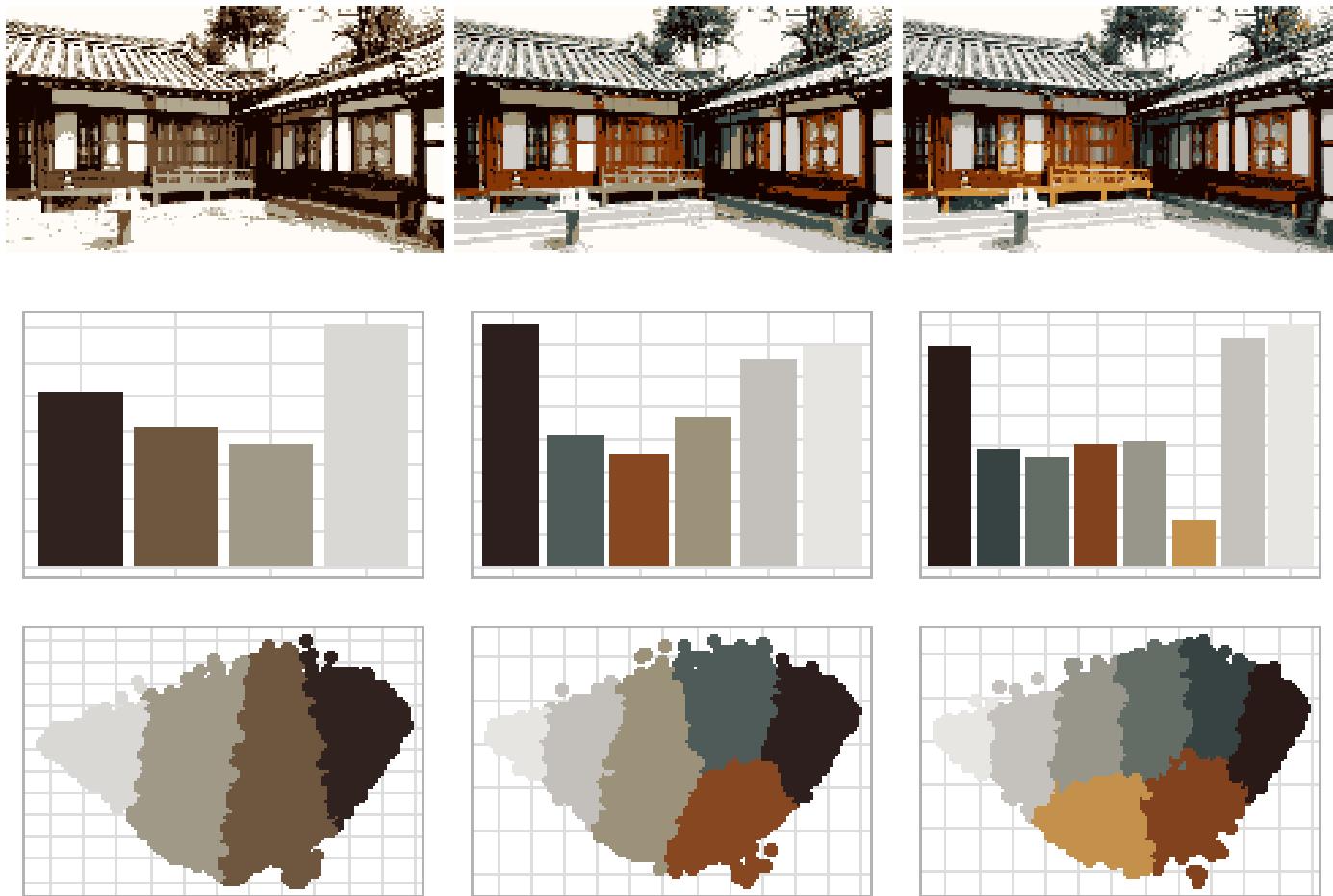
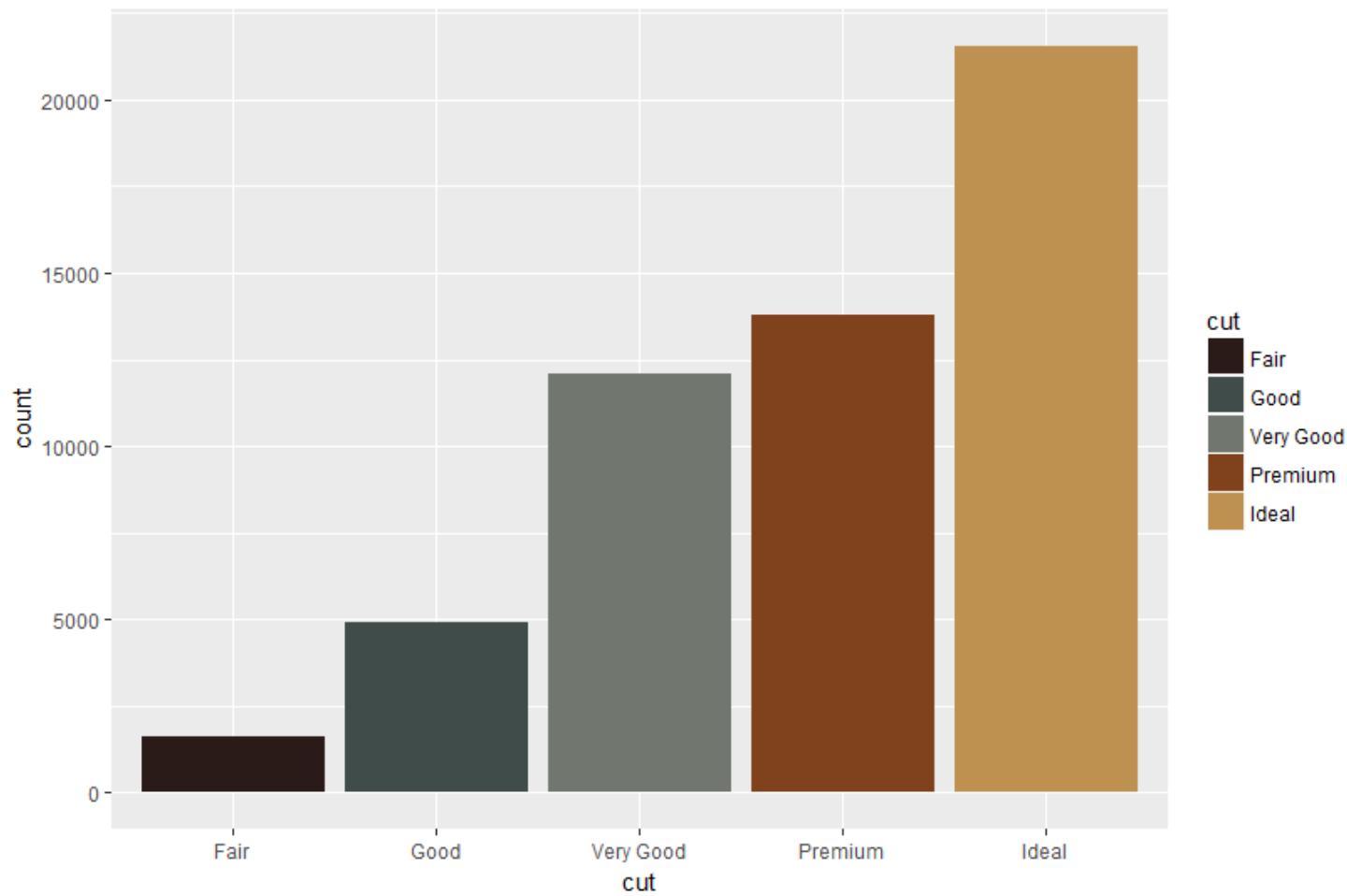


Image to Palette



Image to Palette



Examples



florence1



florence2



santorini 1



amazon



seoul2



Image to Palette

`github.com/shk5660/img2pal`

```
devtools::install_github("shk5660/img2pal")
```

Thanks