

# Color Picker

이덕희

#BCD4EC  
Spindle

#739CEO  
Havelock Blue

#2E5DC2  
Cerulean Blue

#A2854D  
Gold

#33451A  
Thatch Green

---

# 1. 무엇을 하고 싶은지?

---

이미지 속의 특정 부분의 색상을 RGB값으로 추출해 대표 색을 추출하려고 한다.  
특정 이미지 파일에 국한되지 않고 나아가 웹상에 있는 이미지의 URL값을 통해 색을 추출하여  
특정 검색어에 대한 대표 색을 알아보고자 한다.





#BCD4EC  
Spindle

#739CEO  
Havelock Blue

#2E5DC2  
Cerulean Blue

#A2854D  
Gold

#33451A  
Thatch Green

---

## 2. 어떻게 만들었나?

---

### 개발 도구

- Python 3.6.1: Algorithm Implementation
- R 3.4.4 : Interactive Visualization Application

## 2. 어떻게 만들었나?

### 주요 메소드 (PYTHON)



1 Search

- 매개변수(타입): 키워드(string), 페이지(Integer, 100 per 1)
- 리턴값(타입): 이미지 url값(list)

2 Pick\_rgb

- 매개변수(타입): 이미지 url(list)
- 리턴값(타입): 이미지 RGB (dataframe)

3 Color\_kmeans

- 매개변수(타입): 추출할 대표색의 가지 수(integer), 학습 데이터(dataframe)
- 리턴값(타입): 대표 색의 RGB, HEX code(dataframe)

4 Bar\_graph

- 매개변수(타입): RGB(dataframe)
- 리턴값(타입): bar plot, HEX data

## 2. 어떻게 만들었나?

주요 메소드: Search

### Search

```
def search(w,pages):
    global word
    word = w
    url = 'https://www.google.co.kr/'
    ...
    elem.send_keys(w)
    elem.submit()
    ...

    img_url = soup_search.select_one('div > a.q').get('href')
    browser.get(url+img_url)

    for i in range(1,pages):
        browser.find_element_by_tag_name("body").send_keys(Keys.END)
        time.sleep(5)
    ...
    img_list = []
    for i in img_soup:
        t = i.text
        img_list.append(t[t.find('ou')+5:t.find('ow')-3])

    print('Image URL',len(img_list),' is extracted')
    return(img_list)
```

### 주요 라이브러리

```
from selenium import webdriver
from selenium.webdriver.common.keys import Keys
import urllib.request as req
```

- 검색엔진은 구글을 사용
- 매개변수 1 당 100개의 json data 추출
- 슬라이싱을 통해 url list 값 반환

## 2. 어떻게 만들었나?

주요 메소드: Pick\_rgb

### Pick\_rgb

```
def pick_rgb(img_list):
    def converter(url):
        re = req.urlopen(url)
        img = np.array(bytarray(re.read()), dtype='uint8')
        img = cv2.imdecode(img, cv2.IMREAD_COLOR)
        return((img.reshape(-1,3))/256)

    rgb = np.zeros(0).reshape(-1,3)
    ...
    file_type = ['jpg','gif','png','bmp','svg','ico','raw']
    for i in img_list:
        try:
            if i[-3:] in file_type:
                ...
                rgb = np.concatenate([rgb,converter(i)],axis=0)
        except:
            exception.append(i)

    rgb = rgb[((rgb[:,0] != [0] | (rgb[:,1] != [0]) | (rgb[:,2] != [0])))]
    rgb = rgb[((rgb[:,0] != [255] | (rgb[:,1] != [255]) | (rgb[:,2] != [255])))]
    ...
    return(col_df)
```

### 주요 라이브러리

```
import urllib.request as req
Import numpy as np
Import cv
```

- url을 image로 바꾸는 작업이 반복되어 함수 생성
- file\_type에 들어 있지 않는 확장자는 연산 제외
- Youtube, vox 등 영상 화면이 이미지파일로 인식하는 경우 exception 처리
- 사진 배경의 주요 색인 최극단값(흰색, 검은색)은 제외

## 2. 어떻게 만들었나?

주요 메소드: Color\_kmeans

### Kmeans

```
def color_kmeans(n,f):
    model = KMeans(n_clusters=n)
    model.fit(f)
    ...
    centers_rgb = np.zeros(9).reshape(3,3)
    for i in range(3):
        centers_rgb[i][0] = centers[i][2]
        centers_rgb[i][1] = centers[i][1]
        centers_rgb[i][2] = centers[i][0]
    centers = centers_rgb
    ...
    for i in centers:
        colormap.append(mpl.colors.to_hex(i))

    df = pd.DataFrame({'x':np.repeat(1,n),
                       'y':range(1,n+1)})
    ...
    return(df)
```

### 주요 라이브러리

```
from sklearn.cluster import Kmeans
import cv2
Import matplotlib as mpl
```

- openCV의 값은 BGR로 사용하여 순서 변경
- 16진수값인 Hex code 출력
- 슬라이싱을 통해 url list 값 반환
- 그래프를 위한 x,y 축 설정

## 2. 어떻게 만들었나?

주요 메소드: Color\_graph

### Graph

```
def color_graph(df):
    plt.figure(figsize=(10,10))
    plt.barh(df.y,df.x,
             color = df.HEX,
             height = 1)
    plt.axis("off")
    plt.title(word+' color', fontsize = 50)
    plt.xticks()
    plt.yticks()
    plt.annotate(df.HEX[0],xy=(0.3, 1),fontsize = 50,va='center')
    plt.annotate(df.HEX[1],xy=(0.3, 2),fontsize = 50,va='center')
    plt.annotate(df.HEX[2],xy=(0.3, 3),fontsize = 50,va='center')
    plt.show()
```

### 주요 라이브러리

Import matplotlib.pyplot as plt

- 한 줄씩 해당 색 출력
- HEX code를 text형식으로 추출

---

## **2-2.** Starbucks Logo Color Extraction

---

---

시연연상

---

# 시연연상

## 2-2. Starbucks

### Starbucks Image Data

```
,Blue,Green,Red  
0,0.0038909912109375,0.0038909912109375,0.0038909912109375  
1,0.0038909912109375,0.0038909912109375,0.0038909912109375  
2,0.0038909912109375,0.0038909912109375,0.0038909912109375  
3,0.0038909912109375,0.0038909912109375,0.0038909912109375  
4,0.0038909912109375,0.0038909912109375,0.0038909912109375  
...  
...  
7770878,0.0030364990234375,0.00341796875,0.003570556640625  
7770879,0.0030364990234375,0.00341796875,0.003570556640625  
7770880,0.0030364990234375,0.00341796875,0.003570556640625  
7770881,0.0030364990234375,0.00341796875,0.003570556640625  
7770882,0.0030364990234375,0.00341796875,0.003570556640625
```

**총 7,770,458 건 추출**

## 2-2. Starbucks

Starbucks Symbol Color

Starbucks Logo Color

#89b4a0

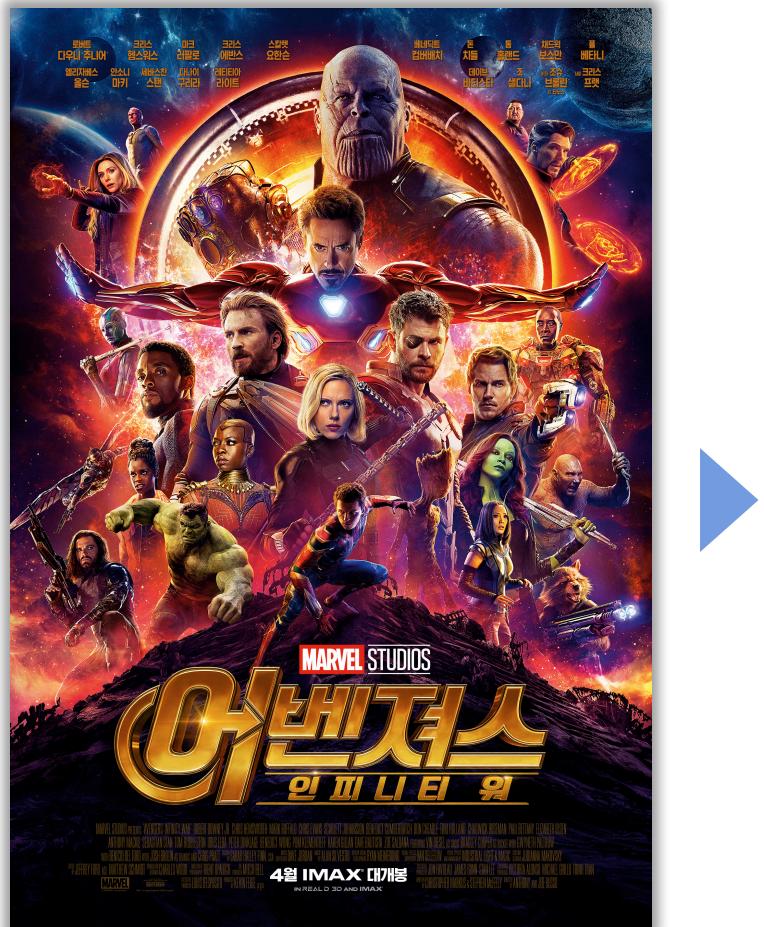
#05693f

#fdfefd

### **3. 추출 결과**

### 3. 추출 결과

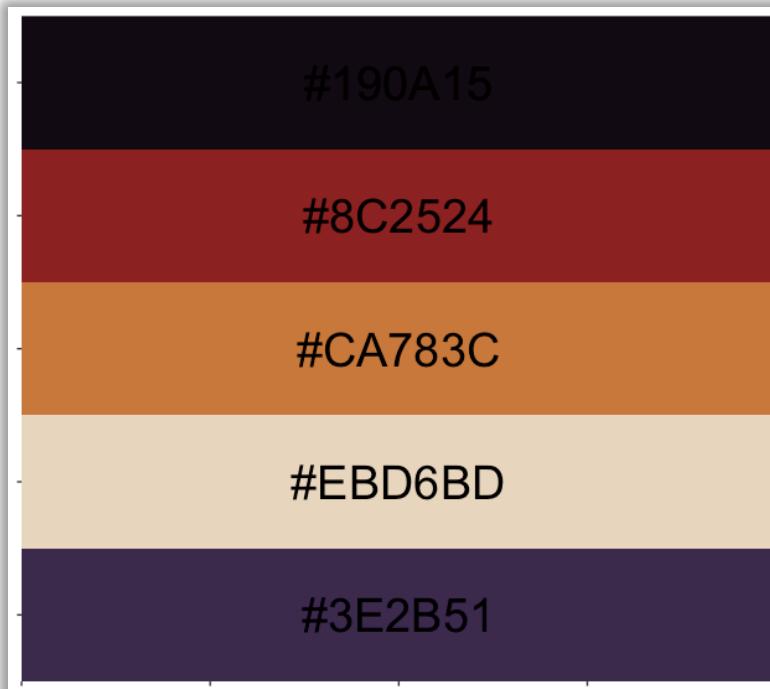
Poster



<Avengers: Infinity War, Marvel, 2018>

### 3. 추출 결과

Poster



<Avengers: Infinity War, Marvel, 2018>

### 3. 추출 결과

Paint



<Monet: Woman with a parasol, 1886>

### 3. 추출 결과

Paint



<Monet: Woman with a parasol, 1886>

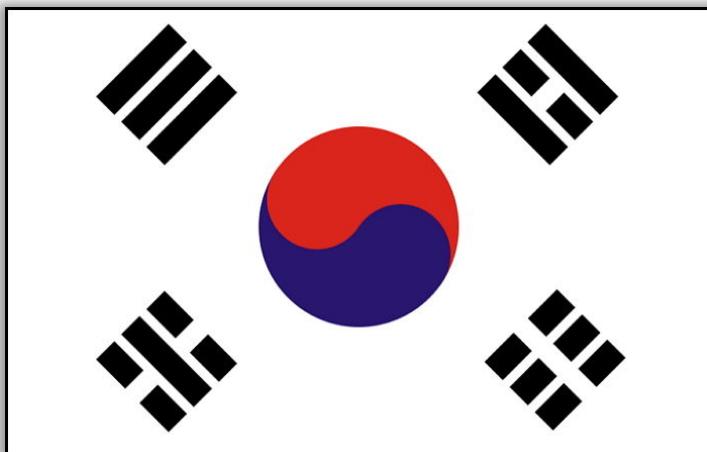
### 3. 추출 결과

Flag



### 3. 추출 결과

Flag



< 태극기 >

---

## 4. Interactive Visualization

---

하나의 파일을 분석하기 위해 Application을 만들어 대표색을 추출해보려 한다. R의 Shiny를 이용하여 이를 만들어 보았다.

---

## 4. 어플리케이션

R with Shiny

---

<http://deokhee.shinyapps.io/colorpicker>

---

# 시연연상

## 5. 활용 방안

---

- 상품 별 고유의 색을 통한 마케팅
- 신상품 개발 시 적합한 색 선정
- 색상 감지로 이상 징후 파악
- 의상, 웹 등의 디자인 톤앤 매너에 활용 등

---

## 6. 보안해야 할 점

---

- 투명도, 밝기 등의 다양한 색상의 속성들을 사용하여 구현
- 색 별 가중치를 통해 중요(특이) 색상 추출
- 구글 외의 검색 엔진을 추가
- 검색어마다 변하는 url 추적
- 코드의 단순화

---

**행복하세요**

---