

別再問我Python可視化如何配色了！

簡說Python 今天

以下文章來源於快學Python



快學Python

Python可視化、自動化辦公、數據分析、爬蟲、Web開發！人生苦短，快學Python！



簡說Python

號主老表，自學，分享Python，SQL零基礎入門、數據分析、數據挖掘、機器學習優質文章以及學習經驗。




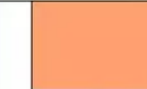
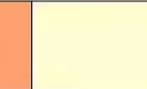



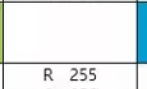

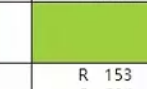
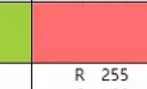
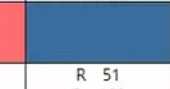

169篇原創內容



公眾號

我們在利用Python進行數據可視化時，有著大量的高質量庫可以用，比如：但圖表好不好看，配色佔一半。如果沒有良好的審美觀，很容易做出來的東西辣眼睛.....

所以想做好數據可視化，就要有合適的配色方案。除了可以藉鑑參考配色網站的案例，也可以自己自定義一套配色方案。

						
R 255 G 102 B 102 #FF6666	R 255 G 255 B 0 #FFFF00	R 0 G 102 B 153 #006699		R 255 G 153 B 102 #FF9966	R 255 G 255 B 204 #FFFFCC	R 0 G 102 B 204 #0066CC
						
R 153 G 204 B 51 #99CC33	R 255 G 255 B 255 #FFFFFF	R 0 G 153 B 204 #0099CC		R 153 G 204 B 51 #99CC33	R 255 G 102 B 102 #FF6666	R 51 G 102 B 153 #336699

如何去自定義呢？

我倒是有一個想法，配色的美感需要培養，但在一開始可以在優秀的作品上尋找靈感，比如經典電影、海報、風景圖、Logo等等，這些都是絕佳的參考。

自然風景的顏色往往令人驚艷，咱們不妨以風景圖為例。下圖是一副海上夕陽圖，通過一番操作就提取到了一套配色方案（見圖右）。





那麼，我們用Python能不能做到呢？

答案當然是可以，畢竟Python除了不能生孩子，什麼都能做！

提取圖片中的配色

在Python中對圖片進行操作，最常用的兩個模塊就是PIL和opencv了。所以一開始我的方案是，用Python庫打開圖片，然後遍歷像素顏色，最後按照色彩比例進行排序，即可得到該圖片的配色方案。



結果做到一半，我發現自己忽略了一件事。大家都知道，Python 是一門優雅的語言，簡潔的語法，強大的功能。同時它還有擁有極其豐富的第三方庫，這些庫幾乎都可以在github 或者pypi上找到源碼。

於是我搜了一下，確實有相關的庫可以提取圖片中的配色，那我們就不用重複造輪子了。

這個模塊就是——

haishoku - A development tool for grabbing the dominant color or representative color palette from

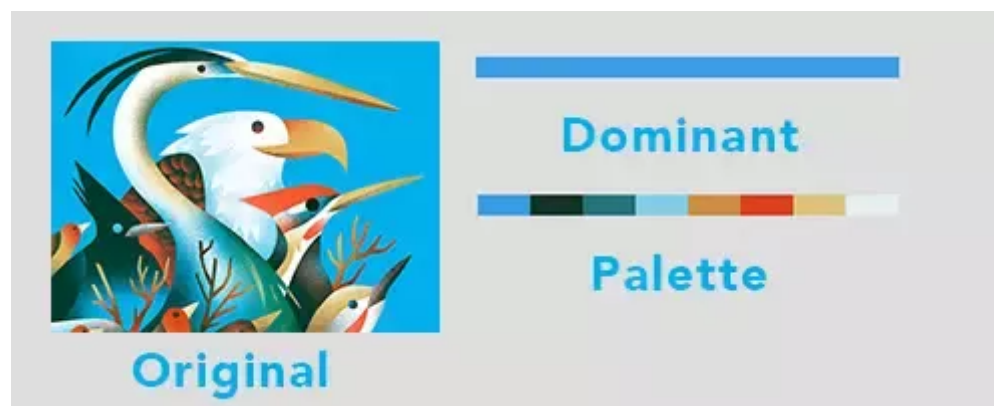
an image



132

haishoku is a JP word, it means palette in English. Haishoku is a development tool for grabbing the dominant color or representative color palette from an image, it depends on Python3 and Pillow.

其GitHub網址為：



具體用法，還是先安裝

```
pip install haishoku
```

將前文提到的海上夕陽圖，保存到本地並命名為 `test.png`

```
from haishoku.haishoku import Haishoku  
image = 'test.png'  
haishoku = Haishoku.loadHaishoku(image)
```

導入模塊，運行代碼會返回一個Haishoku實例，你可以通過實例屬性 `haishoku.dominant` `haishoku.palette`

主色調

首先，要怎麼獲取圖片的主色調呢？

```
print(haishoku.dominant)
```

這返回了一個結構為(R, G, B) 的元組，就是該圖片的主色調。

```
print(haishoku.dominant)  
(217, 208, 190)
```

運行下面這行代碼

```
Haishoku.showDominant(image)
```

則會打開一個臨時文件，用來預覽主色調的顏色。



主色調 (最多的顏色)

配色方案

```
#获取配色方案  
pprint.pprint(haishoku.palette)
```

返回一個結構為：[(R, G, B), (R, G, B), ...] 最大長度為8的數組。

```
#获取配色方案  
pprint.pprint(haishoku.palette)  
  
[(0.42, (217, 208, 190)),  
 (0.15, (211, 137, 124)),  
 (0.12, (132, 115, 126)),  
 (0.11, (245, 192, 141)),  
 (0.07, (156, 177, 189)),  
 (0.06, (9, 12, 56)),  
 (0.05, (129, 162, 183)),  
 (0.02, (49, 60, 100))]
```

這裡使用了 `pprint`

運行下面這行代碼

```
Haishoku.showPalette(image)
```

則會打開一個臨時文件，用來預覽圖片配色方案。（不會保存在本地）



配色方案

就這樣，只需幾行代碼就提取到圖片中的配色方案，是不是很简单。

另外，Haishoku庫從 `v1.1.4`

```
imagepath = 'https://img-blog.csdnimg.cn/20190222215216318.png'  
  
haishoku = Haishoku.loadHaishoku(imagepath)
```

配色方案與可視化

通過前面的操作，我們就提取到了合適的配色，那麼就實戰一下吧。

經典電影、海報、風景圖、Logo都是絕佳的參考對象。



所以這次，我選擇了Google的Logo，並提取到它的配色方案。

```
imagepath = 'google.png'  
  
haishoku = Haishoku.loadHaishoku(imagepath)
```

```
pprint.pprint(haishoku.palette)
```

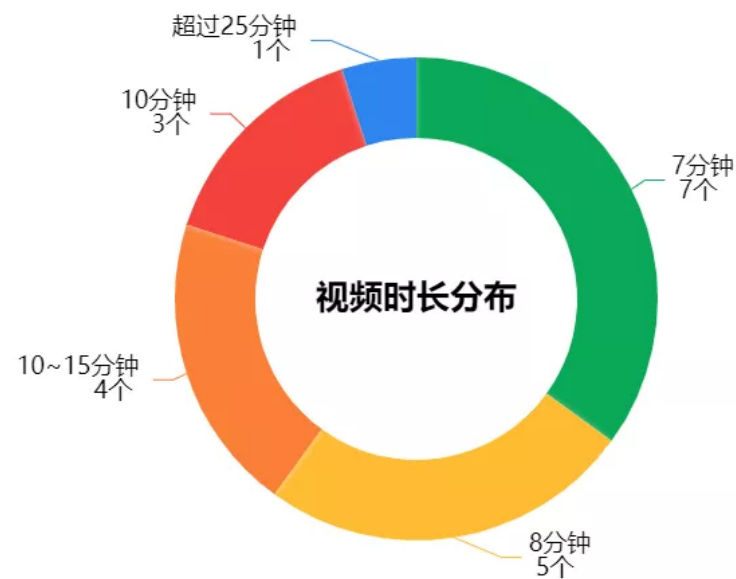
```
Haishoku.showPalette(imagepath)
```

```
[(0.74, (254, 254, 254)),  
(0.12, (66, 134, 244)),  
(0.07, (234, 66, 53)),  
(0.03, (250, 187, 3)),  
(0.02, (49, 167, 81)),  
(0.01, (166, 205, 221)),  
(0.01, (128, 173, 246)),  
(0.0, (250, 196, 37))]
```



那麼，這套配色方案應用到了數據可視化中，會是怎麼樣呢？？

用Python繪製一個餅圖試試看



感覺還不錯，這套配色方案我要收藏起來。如果大家覺得本文還不錯，記得給個一鍵三連！


其實一開始，自己自定義提取圖片配色也不是一點用沒有，我還隨帶用OpenCV製作了一個拾色器。



也就是說，Python運行時只需點擊，就可以獲取當前圖片位置的顏色。


如果大家感興趣的話，就右下角點個贊吧，本文 ！！

圖書推薦：礎操作、圖形處理基本操作、簡單圖形的繪製和對象的管理等內容。



Python爬虫與反爬蟲開發從入門到精通

作者：劉延林

 京東

購買



文章點贊超過100+
我將在個人視頻號直播(
帶大家一起進行項目實戰復現

贈送對應圖書1本



掃碼即可加我微信

老表朋友圈經常有贈書/紅包福利活動

戳我查看简说Python 投稿规则及激励

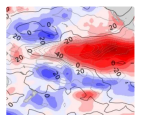
學習更多：

整理了我開始分享學習筆記到現在超過250篇優質文章，涵蓋數據分析、爬蟲、機器學習等方面，別再說不知道該從哪開始，實戰哪裡找了

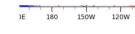


喜歡此內容的人還喜歡

NCAR的計算地球科學Python庫--GeoCAT介紹

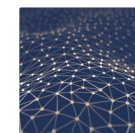


好奇心Log

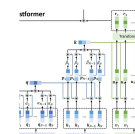


用可視化理解神經網絡！

機器學習實驗室



Fastformer: 簡單又好用的Transformer變體！清華&MSRA開源線性複雜度的Fastformer！



我愛計算機視覺