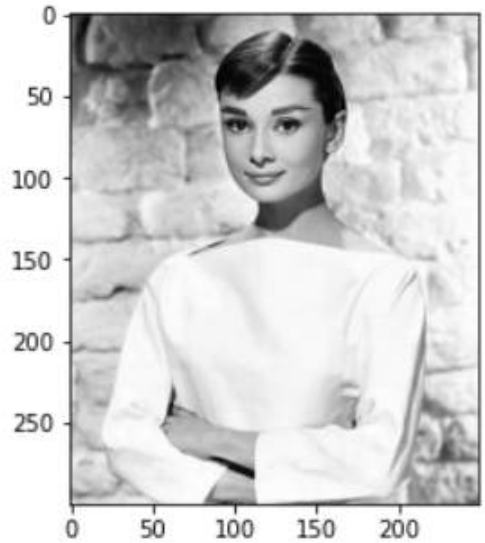
1. selective search

1) 이미지 불러오기

```
import selectivesearch
import cv2
import matplotlib.pyplot as plt
import os
%matplotlib inline

img = cv2.imread('audrey.png')
print(img.shape)
plt.imshow(img)
```

```
(300, 249, 3)
<matplotlib.image.AxesImage at 0x12c81ac7220>
```




```
img_rgb = cv2.cvtColor(img,cv2.COLOR_BGR2RGB)
print(img_rgb.shape)
plt.figure(figsize=(8,8))
# plt.imshow(img_rgb)
```

2) Region Proposal

selectivesearch 패키지의 selective_search를 이용하여 region proposal 실시

```
_, regions = selectivesearch.selective_search(img_rgb)
print(type(regions))
print(len(regions))
```

```
<class 'list'>
698
```

```
regions
```

```
[{'rect': (0, 0, 12, 35), 'size': 200, 'labels': [0.0]},
 {'rect': (10, 0, 9, 29), 'size': 179, 'labels': [1.0]},
 {'rect': (16, 0, 18, 18), 'size': 149, 'labels': [2.0]},
 {'rect': (25, 0, 30, 33), 'size': 395, 'labels': [3.0]},
 {'rect': (44, 0, 25, 17), 'size': 302, 'labels': [4.0]},
 {'rect': (68, 0, 24, 17), 'size': 161, 'labels': [5.0]},
 {'rect': (86, 0, 13, 23), 'size': 165, 'labels': [6.0]},
 {'rect': (94, 0, 12, 24), 'size': 100, 'labels': [7.0]},
 {'rect': (97, 0, 14, 19), 'size': 95, 'labels': [8.0]},
 {'rect': (101, 0, 35, 17), 'size': 358, 'labels': [9.0]},
 {'rect': (134, 0, 13, 14), 'size': 117, 'labels': [10.0]},
 {'rect': (136, 0, 25, 29), 'size': 185, 'labels': [11.0]},
 {'rect': (148, 0, 13, 8), 'size': 96, 'labels': [12.0]},
 {'rect': (147, 0, 26, 21), 'size': 298, 'labels': [13.0]},
 {'rect': (172, 0, 17, 17), 'size': 232, 'labels': [14.0]},
 {'rect': (188, 0, 10, 15), 'size': 136, 'labels': [15.0]},
 {'rect': (196, 0, 21, 15), 'size': 223, 'labels': [16.0]},
 {'rect': (209, 0, 11, 13), 'size': 83, 'labels': [17.0]},
 {'rect': (196, 0, 38, 22), 'size': 234, 'labels': [18.0]},
```

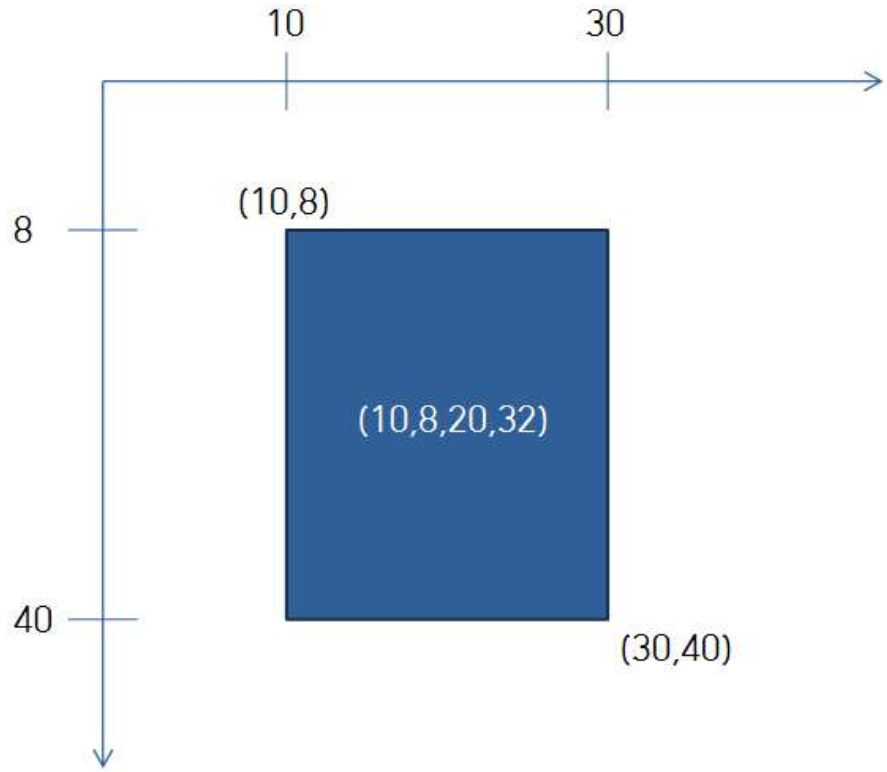
반환된 regions 설명

- 세부 원소는 dictionary 형태의 list이다.
- rect 키 값은 x,y,시작 좌표와 너비 높이값(x,y,w,h)을 가진다.
- rect 키 값은 후보를 나타내는 bounding box
- size는 Object의 크기를 의미한다.
- labels는 해당 rect로 지정된 bounding box 내에 있는 Object들의 고유 ID이다.
- 아래로 내려갈수록 너비와 높이 값이 큰 bounding box이다.
- 아래로 내려갈수록 bounding box 여러 개의 Object가 있을 확률이 증가한다. labels로 해당하는게 많기 때문이다.

```
  201.0]},
 {'rect': (115, 138, 59, 86),
  'size': 2741,
  'labels': [231.0,
   248.0,
   249.0,
   238.0,
   220.0,
   180.0,
   193.0,
   176.0,
   174.0,
   198.0,
   201.0,
   254.0]},
 {'rect': (167, 123, 65, 67),
  'size': 1612,
  'labels': [187.0, 203.0, 207.0, 181.0, 160.0]},
```

3) Bounding box 시각화

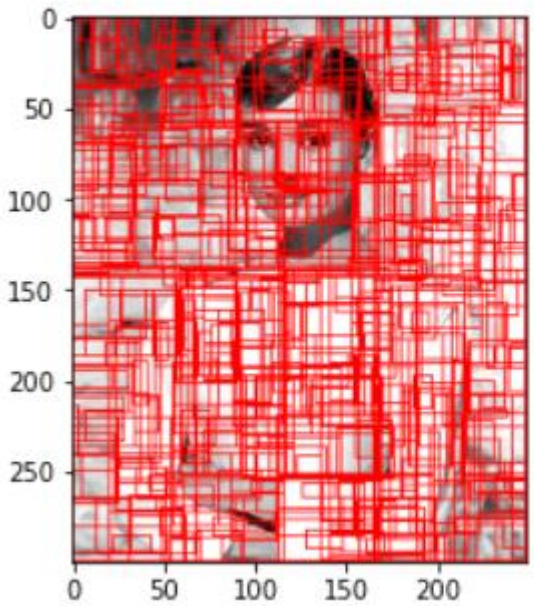
```
#rect만 따로 뽑기
rect = []
for i in range(len(regions)):
    rect.append(regions[i]['rect'])
```



```
blue = (255,0,0)
img_rgb_copy = img_rgb.copy()


for i in range(len(rect)):
    left = rect[i][0]
    top = rect[i][1]
    right = left + rect[i][2]
    bottom = top + rect[i][3]
    # 0은 thickness이다.
    img_rgb_copy = cv2.rectangle(img_rgb_copy,(left,top),(right,bottom),blue,0)
plt.imshow(img_rgb_copy)
```

```
<matplotlib.image.AxesImage at 0x12c8206ca30>
```



의문) blue box를 그렸는데 왜 red box가 나왔지?
답변) 아까 앞에서 그림을 RGB로 정의를 해줬기 때문에 R = 255라 red 가 나왔다.

```
# bounding box 크기가 큰 후보만 추출
size_over_10000 = [i['rect'] for i in regions if i['size']>10000]
size_over_10000
```

```
[(20, 58, 228, 241),
 (0, 0, 189, 143),
 (157, 0, 91, 299),
 (141, 0, 107, 299),
 (0, 99, 116, 200),
 (141, 0, 107, 299),
 (0, 85, 116, 214),
 (0, 85, 116, 214),
 (0, 0, 189, 143),
 (0, 0, 189, 299),
 (141, 0, 107, 299),
 (141, 0, 107, 299),
 (141, 0, 107, 299),
 (141, 0, 107, 299),
 (141, 0, 107, 299),
 (136, 0, 112, 299),
 (136, 0, 112, 299),
 (20, 58, 228, 241),
 (136, 0, 112, 299),
 (136, 0, 112, 299),
 (0, 0, 248, 299),
 (20, 58, 228, 241),
 (20, 58, 228, 241),
 (20, 58, 228, 241),
 (0, 0, 248, 299),
 (0, 0, 248, 299),
 (0, 0, 248, 299),
 (0, 0, 248, 299),
 (0, 0, 248, 299),
 (0, 0, 248, 299),
 (0, 0, 248, 299),
 (0, 0, 248, 299)]
```

이를 시각화하면 다음과 같다.

```
img_rgb_copy2 = img_rgb.copy()



for i in range(len(size_over_10000)):
    left = size_over_10000[i][0]
    top = size_over_10000[i][1]
    right = left + size_over_10000[i][2]
    bottom = top + size_over_10000[i][3]

    img_rgb_copy2 = cv2.rectangle(img_rgb_copy2,(left,top),(right,bottom),blue,0)
```

```
plt.imshow(img_rgb_copy2)
```

```
<matplotlib.image.AxesImage at 0x12c822a5af0>
```