

## 4. Image Quality

Code path: *mutu/Samples/OpenCV/01\_Getting\_Started\_with\_OpenCV/03\_OpenCV Picture quality.ipynb*

1. Compression method.

```
cv2.imwrite('yahboomTest.jpg', img, [cv2.IMWRITE_JPEG_QUALITY, 50])
```

cv2.CV\_IMWRITE\_JPEG\_QUALITY: Set the image quality of the image format to .jpeg or .jpg, the value is 0---100 (the larger the value, the higher the quality), the default is 95

cv2.CV\_IMWRITE\_WEBP\_QUALITY: Set the image format to .webp format image quality, the value is 0--100

cv2.CV\_IMWRITE\_PNG\_COMPRESSION: Set the compression ratio of .png format, the value is 0--9 (the larger the value, the greater the compression ratio), the default is 3

The main code is as follows:

```
import cv2
img = cv2.imread('yahboom.jpg',1)
cv2.imwrite('yahboomTest.jpg', img, [cv2.IMWRITE_JPEG_QUALITY, 50])
#1M 100k 10k 0-100 Lossy compression
```

```
# 1 Lossless 2 Transparency Property
import cv2
img = cv2.imread('yahboom.jpg',1)
cv2.imwrite('yahboomTest.png', img, [cv2.IMWRITE_PNG_COMPRESSION,0])
# jpg 0 Compression ratio 0-100 higher    png 0 Compression ratio 0-9 lower
```

```
#bgr8 to jpeg format#bgr8 to jpeg format
import enum
import cv2
def bgr8_to_jpeg(value, quality=75):
    return bytes(cv2.imencode('.jpg', value)[1])
```

```
import ipywidgets.widgets as widgets
image_widget1 = widgets.Image(format='jpg', )
image_widget2 = widgets.Image(format='jpg', )
# create a horizontal box container to place the image widget next to
eachother
image_container = widgets.HBox([image_widget1, image_widget2])
# display the container in this cell's output
display(image_container)
img1 = cv2.imread('yahboomTest.jpg',1)
img2 = cv2.imread('yahboomTest.png',1)
image_widget1.value = bgr8_to_jpeg(img1)
image_widget2.value = bgr8_to_jpeg(img2)
```

When the code block runs to the end, you can see the comparison of the two photos.

