OpenCV Resize image using cv2.resize()

Resizing an image means changing the dimensions of it, be it width alone, height alone or both. Also, the aspect ratio of the original image could be preserved in the resized image. To resize an image, OpenCV provides cv2.resize() function.

In this tutorial, we shall the syntax of cv2.resize and get hands-on with examples provided for most of the scenarios encountered in regular usage.

Syntax of cv2.resize()

Following is the syntax of resize function in OpenCV:

```
cv2.resize(src, dsize[, dst[, fx[, fy[, interpolation]]]])
```

The description about the parameters of resize function.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>src</td>
<td>[required] source/input image</td>
</tr>
<tr>
<td>dsize</td>
<td>[required] desired size for the output image</td>
</tr>
<tr>
<td>fx</td>
<td>[optional] scale factor along the horizontal axis</td>
</tr>
<tr>
<td>fy</td>
<td>[optional] scale factor along the vertical axis</td>
</tr>
<tr>
<td>interpolation</td>
<td>[optional] flag that takes one of the following methods. INTER_NEAREST – a nearest-neighbor interpolation INTER_LINEAR – a bilinear interpolation (used by default) INTER_AREA – resampling using pixel area relation. It may be a preferred method for image decimation, as it gives moire’-free results. But when the image is zoomed, it is similar to the INTER_NEAREST method. INTER_CUBIC – a bicubic interpolation over 4×4 pixel neighborhood INTER_LANCZOS4 – a Lanczos interpolation over 8×8 pixel neighborhood</td>
</tr>
</tbody>
</table>

Examples of using cv2.resize() function

Resizing an image can be done in many ways. We will look into examples demonstrating the following resize operations.

1. Preserve Aspect Ratio (height to width ratio of image is preserved)
   1. Downscale (Decrease the size of the image)
   2. Upscale (Increase the size of the image)
2. Do not preserve Aspect Ratio
   1. Resize only the width (Increase or decrease the width of the image keeping height unchanged)
   2. Resize only the height (Increase or decrease the height of the image keeping width unchanged)
3. Resize to specific width and height

Following is the original image with dimensions (149,200,4) (height, width, number of channels) on which we shall experiment on:

![Image](image.png)

**Preserve Aspect Ratio**

**Downscale with resize()**

In the following example, `scale_percent` value holds the percentage by which image has to be scaled. Providing a value <100 downscales the image provided. We will use this `scale_percent` value along with original image's dimensions to calculate the width and height of output image.

```python
import cv2

img = cv2.imread('/home/img/python.png', cv2.IMREAD_UNCHANGED)
print('Original Dimensions : ',img.shape)

scale_percent = 60 # percent of original size
width = int(img.shape[1] * scale_percent / 100)
height = int(img.shape[0] * scale_percent / 100)
dim = (width, height)
# resize image
resized = cv2.resize(img, dim, interpolation = cv2.INTER_AREA)
print('Resized Dimensions : ',resized.shape)

cv2.imshow("Resized image", resized)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

**Output**

```
Original Dimensions : (149, 200, 4)
Resized Dimensions : (89, 120, 4)
```
The original image with dimensions $[149 \times 200 \times 4]$ has been resized to $[89, 120, 4]$ using resize() function.

**Upscale with resize()**

In the following example, scale_percent value holds the percentage by which image has to be scaled. Providing a value >100 upscales the image provided.

```python
import cv2

# load the image
img = cv2.imread('/home/img/python.png', cv2.IMREAD_UNCHANGED)

print('Original Dimensions : ',img.shape)

scale_percent = 220  # percent of original size
width = int(img.shape[1] * scale_percent / 100)
height = int(img.shape[0] * scale_percent / 100)
dim = (width, height)

# resize image
resized = cv2.resize(img, dim, interpolation = cv2.INTER_AREA)

print('Resized Dimensions : ',resized.shape)

cv2.imshow("Resized image", resized)
cv2.waitKey(0)
cv2.destroyAllWindows()""
```

Output

```
Original Dimensions : (149, 200, 4)
Resized Dimensions : (327, 440, 4)
```
Do not preserve the aspect ratio

Resize only width

In this example, we provided a specific value in pixels for width and left the height unchanged.

```python
import cv2
```
import cv2

img = cv2.imread('/home/img/python.png')

print('Original Dimensions : ', img.shape)

width = 440
height = img.shape[0] # keep original height
dim = (width, height)

# resize image
resized = cv2.resize(img, dim, interpolation = cv2.INTER_AREA)

print('Resized Dimensions : ', resized.shape)

cv2.imshow("Resized image", resized)
cv2.waitKey(0)
cv2.destroyAllWindows()

---

As we have increased only the width, the output image looks stretched horizontally.

---

Resize only height

In the following example, scale_percent value holds the percentage by which height has to be scaled. Or you may also provide a specific value in pixels.

import cv2
import cv2

img = cv2.imread('/home/img/python.png', cv2.IMREAD_UNCHANGED)

print('Original Dimensions : ', img.shape)

width = img.shape[1]  # keep original width
height = 440
dim = (width, height)

# resize image
resized = cv2.resize(img, dim, interpolation = cv2.INTER_AREA)

print('Resized Dimensions : ', resized.shape)

cv2.imshow("Resized image", resized)
cv2.waitKey(0)
cv2.destroyAllWindows()

Output

As we have increased only the height, the output image looks stretched vertically.

Resize to specific width and height
In the following example, we shall provide specific value in pixels for both width and height.

```python
import cv2

img = cv2.imread('/home/img/python.png', cv2.IMREAD_UNCHANGED)

print('Original Dimensions : ', img.shape)

width = 350
height = 450
dim = (width, height)

# resize image
resized = cv2.resize(img, dim, interpolation = cv2.INTER_AREA)

print('Resized Dimensions : ', resized.shape)

cv2.imshow("Resized image", resized)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

Output

```
Original Dimensions : (149, 200, 4)
Resized Dimensions : (450, 350, 4)
```
Concluding this tutorial, we have learned how to resize an image in Python using OpenCV `resize()` function.