jdhzoqbdy

April 21, 2023

Topics:

- Reading an image file and converting it to a numpy array
- Resizing an image
- RGB to Grayscale conversion

[ ]:

```
# getting an image using web get

!wget 'https://tractive.com/blog/wp-content/uploads/2016/04/
 ↪puppy-care-guide-for-new-parents.jpg'
```

```
--2022-03-17 15:37:15--  https://tractive.com/blog/wp-
content/uploads/2016/04/puppy-care-guide-for-new-parents.jpg
Resolving tractive.com (tractive.com)… 54.192.18.89, 54.192.18.83,
54.192.18.105, …
Connecting to tractive.com (tractive.com)|54.192.18.89|:443… connected.
HTTP request sent, awaiting response… 200 OK
Length: 113570 (111K) [image/jpeg]
Saving to: ‘puppy-care-guide-for-new-parents.jpg’

puppy-care-guide-fo 100%[===================>] 110.91K   382KB/s    in 0.3s

2022-03-17 15:37:16 (382 KB/s) - ‘puppy-care-guide-for-new-parents.jpg’ saved
[113570/113570]
```

Libraries that can be used for image processing:

1. matplotlib.image
2. Pillow
3. OpenCV (cv2)

[ ]:

```
# importing the image module from matplotlib library

import matplotlib.image as mpimg
import matplotlib.pyplot as plt
```

[ ]:

```python
# loading an image through matplotlib.image module

img = mpimg.imread('/content/dog.jpg')
```

[ ]:

```python
type(img)
```

[ ]: numpy.ndarray

[ ]:

```python
print(img.shape)
```

```
(1365, 2048, 3)
```

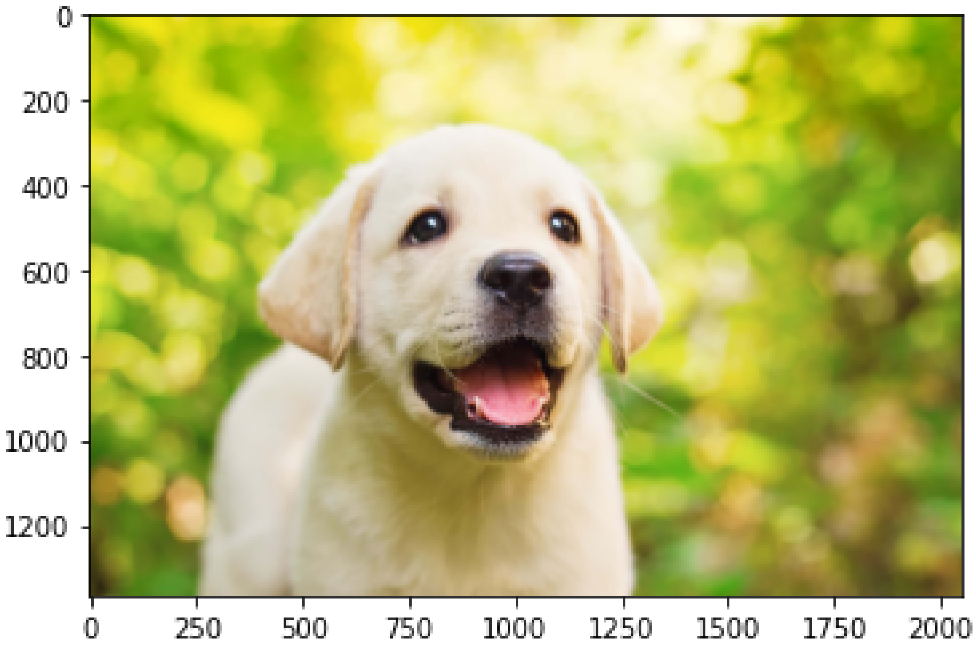
[ ]:

```python
print(img)
```

```
[[[147 182   0]
  [147 182   0]
  [147 182   0]
  …
  [128 148   0]
  [128 148   0]
  [128 148   0]]

 [[146 181   0]
  [146 181   0]
  [147 182   0]
  …
  [128 148   0]
  [128 148   0]
  [128 148   0]]

 [[145 180   0]
  [145 180   0]
  [145 180   0]
  …
  [128 148   0]
  [128 148   0]
  [128 148   0]]

 …

 [[ 76  69  25]
  [ 76  69  25]
  [ 75  68  24]
  …
  [135 106  38]
  [135 106  38]
  [135 106  38]]
 [[ 76  69  25]
  [ 76  69  25]
  [ 75  68  24]
  …
  [135 106  38]
  [135 106  38]
  [135 106  38]]

 [[ 76  69  25]
  [ 76  69  25]
  [ 75  68  24]
  …
  [135 106  38]
  [135 106  38]
  [135 106  38]]]
```

[ ]:

```
# displaying the image from numpy array

img_plot = plt.imshow(img)
plt.show()
```



Resizing the image using Pillow library

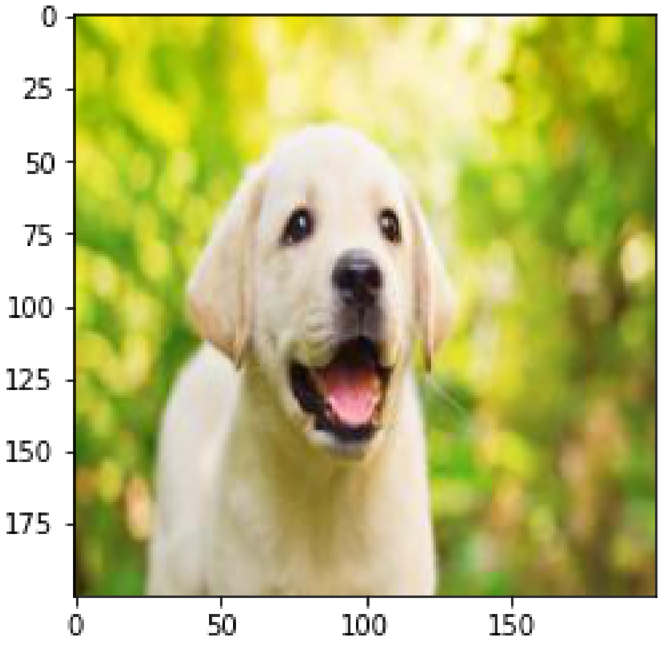
[ ]:

```
from PIL import Image
```

```
[ ]: img = Image.open('/content/dog.jpg')
     img_resized = img.resize((200, 200))
```

```
[ ]: img_resized.save('dog_image_resized.jpg')
```

```
[ ]: # displaying the image from numpy array

     img_res = mpimg.imread('/content/dog_image_resized.jpg')
     img_res_plot = plt.imshow(img_res)
     plt.show()
```



```
[ ]: print(img_res.shape)
```

```
(200, 200, 3)
```

Converting RGB images to Grayscale image using OpenCV

```
[ ]: # importing OpenCV library
     import cv2
```

```
[ ]: img = cv2.imread('/content/dog.jpg')
```

```
[ ]: type(img)
```

```
[ ]: numpy.ndarray
[ ]: img.shape
```

```
[ ]: (1365, 2048, 3)
```

```
[ ]: grayscale_image = cv2.cvtColor(img, cv2.COLOR_RGB2GRAY)
```

```
[ ]: type(grayscale_image)
```

```
[ ]: numpy.ndarray
```

```
[ ]: grayscale_image.shape
```

```
[ ]: (1365, 2048)
```

cv2.imshow() will display the image. But this will not be allowed in Google Colab.

from google.colab.patches import cv2__imshow

```
[ ]: from google.colab.patches import cv2_imshow
```

```
[ ]: # displaying the image
     cv2_imshow(grayscale_image)
```

[ ]:

```
# saving the grayscale image
cv2.imwrite('dog_grayscale_image.jpg', grayscale_image)
```

[ ]:

```
True
```

[ ]: