

Image data processing

Process involve -

- 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://cdn.britannica.com/55/174255-050-526314B6/brown-Guernsey-cow.jpg'
```

```
--2024-12-20 11:45:17-- https://cdn.britannica.com/55/174255-050-526314B6/brown-Guernsey-cow.jpg
Resolving cdn.britannica.com (cdn.britannica.com)... 3.169.137.83, 3.169.137.2, 3.169.137.82, ...
Connecting to cdn.britannica.com (cdn.britannica.com)|3.169.137.83|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 314174 (307K) [image/jpeg]
Saving to: 'brown-Guernsey-cow.jpg'

brown-Guernsey-cow. 100%[=====>] 306.81K  720KB/s   in 0.4s

2024-12-20 11:45:18 (720 KB/s) - 'brown-Guernsey-cow.jpg' saved [314174/314174]
```

Libraries that we use 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
```

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

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

```
type(img)
```

```
numpy.ndarray
```

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

```
(1064, 1600, 3)
```

```
print(img)
```

```
[[[167 192 223]
    [167 192 223]
    [167 192 223]
    ...
    [222 240 254]
    [222 240 254]
    [222 240 254]]]
```

```
[166 191 222]
[166 191 222]
[166 191 222]
...
[221 239 253]
[221 239 253]
[221 239 253]]
```

```
[165 190 221]
[165 190 221]
[165 190 221]
...
[220 238 252]
[219 237 251]
[219 237 251]]
```

```
...
```

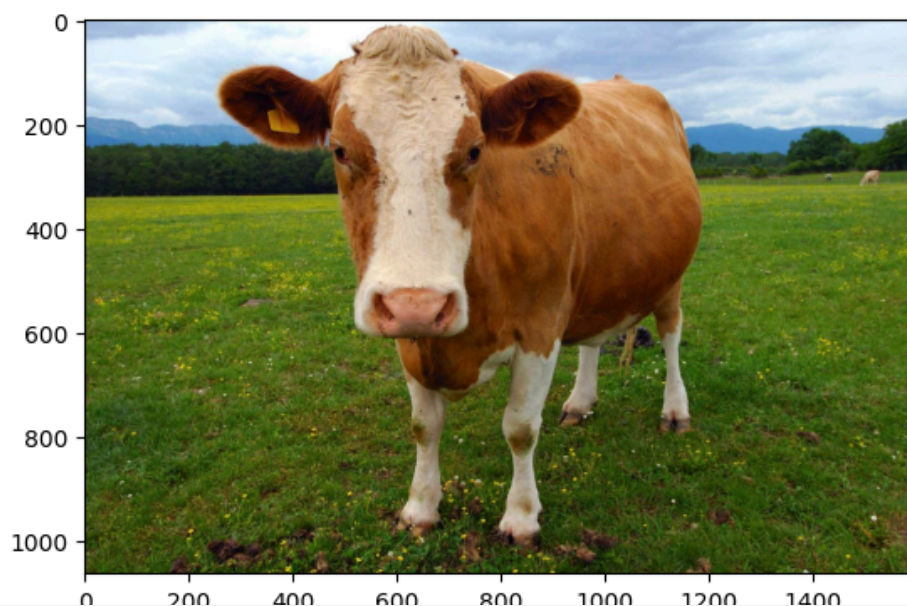
```
[ [ 37 34 0]
  [ 24 24 0]
  [ 52 58 12]
  ...
  [ 61 68 24]
  [ 48 50 10]
  [ 60 60 22]]
```

```
[ [ 37 34 1]
  [ 44 44 6]
  [ 61 67 23]
  ...
  [ 56 63 19]
  [ 50 52 13]
  [ 73 73 37]]
```

```
[ [ 56 53 20]
  [ 78 78 42]
  [ 70 76 32]
  ...
  [ 59 66 22]
  [ 33 35 0]
  [ 42 42 6]]]
```

```
# displaying the image from numpy array
```

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



Resizing the image using Pillow library

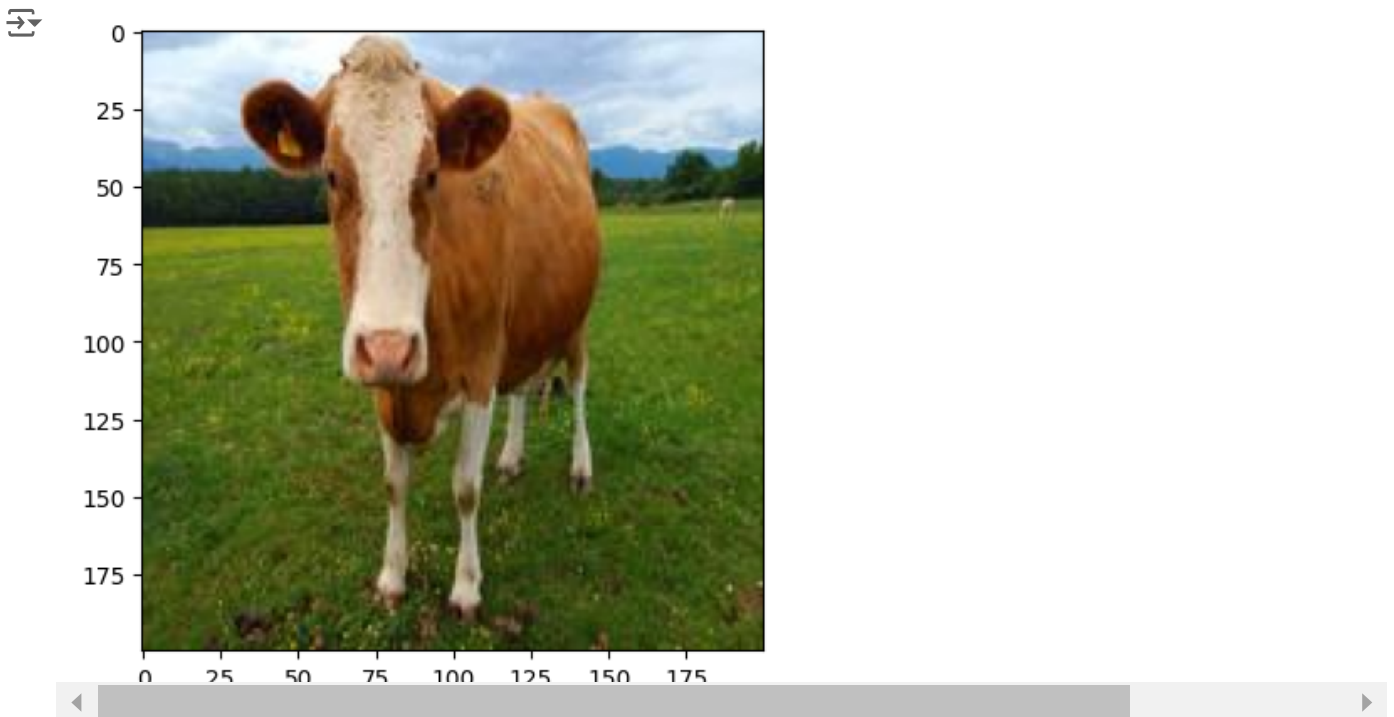
```
from PIL import Image

img = Image.open('/content/cow.jpg')
img_resized = img.resize((200, 200))

img_resized.save('cow_image_resized.jpg')

# displaying the image from numpy array

img_res = mpimg.imread('/content/cow_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/cow.jpg')
```

```
type(img)
```

```
numpy.ndarray
```

```
img.shape
```

```
(1064, 1600, 3)
```

```
grayscale_image = cv2.cvtColor(img, cv2.COLOR_RGB2GRAY)
```

```
type(grayscale_image)
```

```
→ numpy.ndarray
```

```
grayscale_image.shape
```

```
→ (1064, 1600)
```

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('cow_grayscale_image.jpg', grayscale_image)
```

```
→ True
```

Start coding or [generate](#) with AI.

