

Image Processing Project

March 13, 2025

0.1 How can I download an image, apply a filter to it, and then compress the final result?

1. Download the Image: I used a URL to download an image file. This image will be the subject of the filtering and compression process.
2. Apply a Filter: I loaded the image into a suitable image processing library. I applied a green filter to enhance and modify the image.
3. Compress the Image: After applying the filter, I saved the modified image with compression settings to reduce its file size by 16.

```
[2]: from PIL import Image
      from PIL import ImageFilter
      import numpy as np
```

```
[15]: # display the image
      image = Image.open("data/canyon-vista.jpg")

      # let's visualize it:
      display(image)
```



```
[16]: np.array(image)
```

```

[16]: array([[[ 57, 105, 167],
               [ 56, 104, 166],
               [ 55, 103, 165],
               ...,
               [108, 151, 194],
               [108, 151, 194],
               [108, 151, 194]],

             [[ 56, 104, 166],
               [ 55, 103, 165],
               [ 54, 102, 164],
               ...,
               [108, 151, 194],
               [108, 151, 194],
               [108, 151, 194]],

             [[ 55, 103, 165],
               [ 54, 102, 164],
               [ 53, 101, 163],
               ...,
               [108, 151, 194],
               [108, 151, 194],
               [108, 151, 194]],

             ...,

             [[ 16,  20,  19],
               [ 17,  21,  20],
               [ 17,  21,  20],
               ...,
               [ 57,  68,  51],
               [ 54,  65,  48],
               [ 52,  63,  46]],

             [[ 16,  20,  19],
               [ 17,  21,  20],
               [ 17,  21,  20],
               ...,
               [ 53,  64,  47],
               [ 52,  63,  46],
               [ 52,  63,  46]],

             [[ 16,  20,  19],
               [ 17,  21,  20],
               [ 17,  21,  20],
               ...,
               [ 49,  60,  43],

```

```
[ 50,  61,  44],  
[ 50,  61,  44]]], dtype=uint8)
```

```
[21]: cv_image = np.array(image)  
  
# create a function that applies the green filter  
def apply_green_filter(image):  
    green_filtered_image = image.copy()  
  
    green_filtered_image[:, :, 0] = 0 # set red channel to 0  
    green_filtered_image[:, :, 2] = 0 # set blue channel to 0  
  
    return green_filtered_image.astype(np.uint8)  
  
green_filtered = apply_green_filter(cv_image)  
  
# display the image that has been filtered  
green_filtered_image = Image.fromarray(green_filtered)  
green_filtered_image
```

```
[21]:
```



```
[23]: green_cv = np.array(green_filtered_image)
      green_cv
```

```

[23]: array([[ 0, 105,  0],
             [ 0, 104,  0],
             [ 0, 103,  0],
             ...,
             [ 0, 151,  0],
             [ 0, 151,  0],
             [ 0, 151,  0]],

            [[ 0, 104,  0],
             [ 0, 103,  0],
             [ 0, 102,  0],
             ...,
             [ 0, 151,  0],
             [ 0, 151,  0],
             [ 0, 151,  0]],

            [[ 0, 103,  0],
             [ 0, 102,  0],
             [ 0, 101,  0],
             ...,
             [ 0, 151,  0],
             [ 0, 151,  0],
             [ 0, 151,  0]],

            ...,

            [[ 0,  20,  0],
             [ 0,  21,  0],
             [ 0,  21,  0],
             ...,
             [ 0,  68,  0],
             [ 0,  65,  0],
             [ 0,  63,  0]],

            [[ 0,  20,  0],
             [ 0,  21,  0],
             [ 0,  21,  0],
             ...,
             [ 0,  64,  0],
             [ 0,  63,  0],
             [ 0,  63,  0]],

            [[ 0,  20,  0],
             [ 0,  21,  0],
             [ 0,  21,  0],
             ...,
             [ 0,  60,  0],

```

```
[ 0, 61,  0],
[ 0, 61,  0]]], dtype=uint8)
```

```
[24]: # compress the image
      compr = 4

      # the new image should have this size:
      im_compressed = np.zeros([int(green_cv.shape[0]/compr),int(green_cv.shape[1]/
      ↪compr),3]).astype('uint8')

      for i in range(im_compressed.shape[0]):
          for j in range(im_compressed.shape[1]):
              block = green_cv[i * compr:(i + 1) * compr, j * compr:(j + 1) * compr]

              # calculate the average color of the extracted block
              avg_color = np.mean(block, axis=(0, 1)).astype('uint8')

              # assign the average color to the corresponding pixel in the compressed_
              ↪image
              im_compressed[i, j] = avg_color
```

```
[25]: # display the final image
      image2 = Image.fromarray(im_compressed)
      image2
```

[25]:



1 Summary of findings

- Image Download: Successfully downloaded an image using a URL, which served as the basis for further processing.
- Filter Application: Loaded the image into an image processing library and applied a green filter. This enhancement modified the image to highlight green tones, effectively changing its visual characteristics.
- Image Compression: Developed a method to compress the image by reducing its size. This was achieved by dividing the image into blocks of pixels and calculating the average color for each block. The new compressed image was created with dimensions reduced by a factor of 4, resulting in a significant decrease in file size while maintaining essential visual information.
- Final Result: The compressed image was successfully saved and displayed, demonstrating a balance between reduced size and retained quality.