img

February 20, 2025

[11]:
```
import numpy as np
import matplotlib.pyplot as plt


img = plt.imread("kodim23.png")
```

[12]:
```
plt.imshow(img)
```

[12]: <matplotlib.image.AxesImage at 0x7d0d37eeec90>

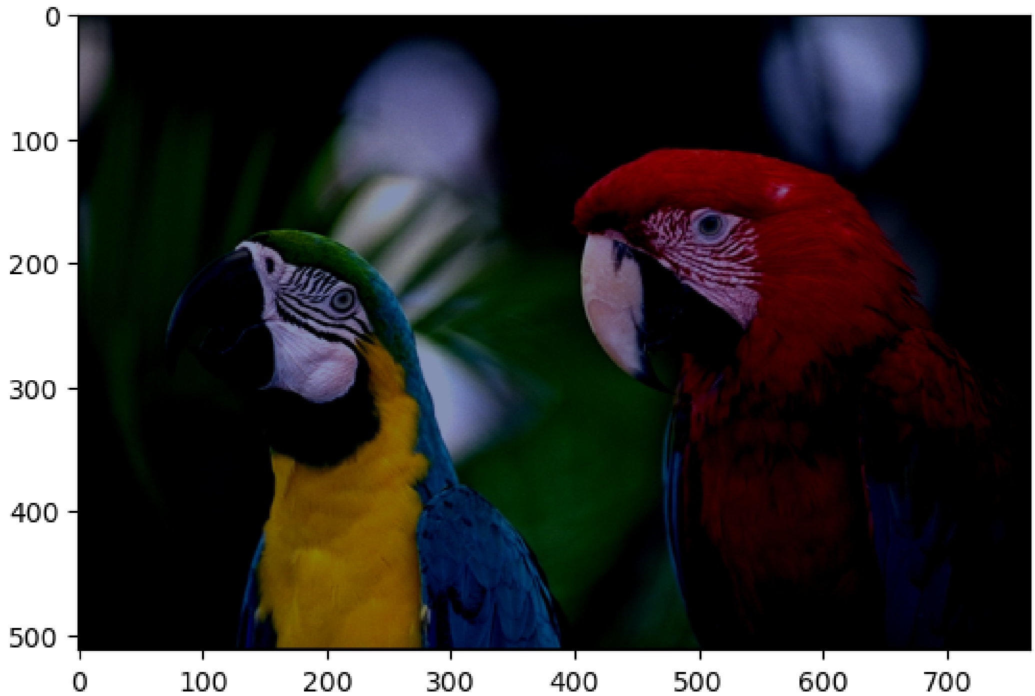
[13]:
```
#Part A and B
means = img.mean((0,1))

zero_mean = img - means
```

```
plt.imshow(zero_mean)
```

```
Clipping input data to the valid range for imshow with RGB data ([0..1] for
floats or [0..255] for integers). Got range [-0.4771625..0.70240855].
```

[13]:
```
<matplotlib.image.AxesImage at 0x7d0d37d6d880>
```



[14]:
```
#Part c
#Flattening out the channels
img_flat = zero_mean.reshape((-1,3))


cov_matrix = np.cov(img_flat,rowvar=0)

cov_matrix
```

[14]:
```
array([[0.05075944, 0.02759223, 0.02004322],
       [0.02759223, 0.03968848, 0.02683978],
       [0.02004322, 0.02683978, 0.04420699]])
```

[15]:
```
#Parts d,e,f
eigvals,eigvecs = np.linalg.eig(cov_matrix)
```

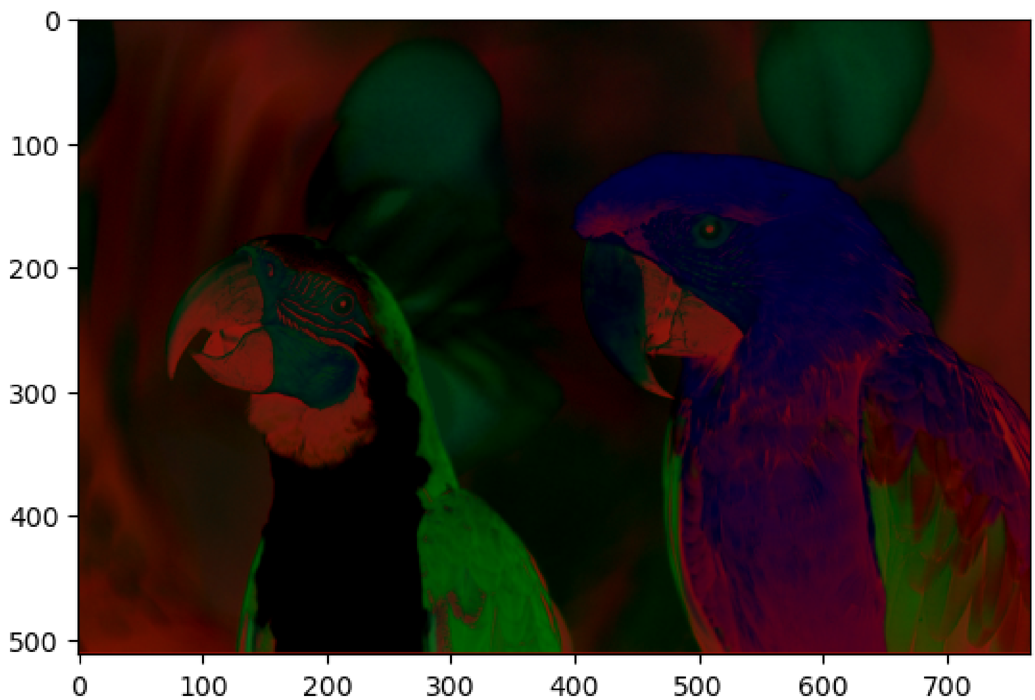
```
print(eigvals,eigvecs)
kl_transformed = np.dot(img_flat,eigvecs).reshape(img.shape)

plt.imshow(kl_transformed)
```

```
Clipping input data to the valid range for imshow with RGB data ([0..1] for
floats or [0..255] for integers). Got range
[-1.030023928616075..0.7003015767285393].
```

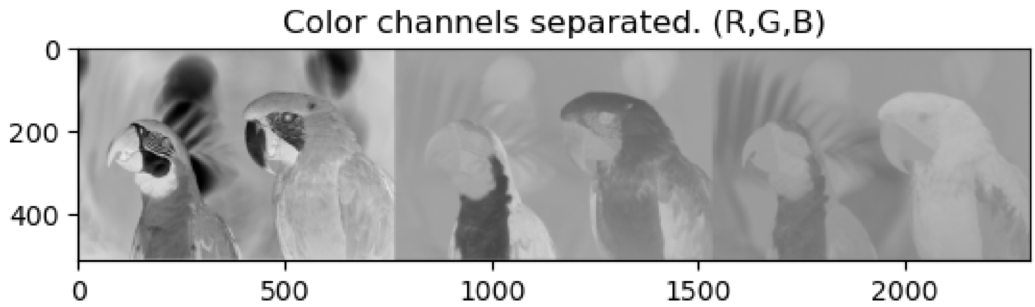
```
[0.09466897 0.02744862 0.01253732] [[-0.60995797 -0.71697239  0.33749352]
 [-0.57321392  0.1051337  -0.8126332 ]
 [-0.54715362  0.68912807  0.47510568]]
```

[15]: <matplotlib.image.AxesImage at 0x7d0d37ff96a0>



[16]:
```
#Part g
channels = np.dsplit(kl_transformed,3)
# channels = [np.pad(image.squeeze(),10) for image in channels]
images = np.hstack(channels)
plt.title("Color channels separated. (R,G,B)")
plt.imshow(images,cmap="gray")
```

[16]: <matplotlib.image.AxesImage at 0x7d0d37fd0920>



[17]:
```
#Part h
cov_matrix_2 = np.cov(kl_transformed.reshape(-1,3),rowvar=0)
cov_matrix_2

print("Covariance matrix:\n",cov_matrix_2,"\n")

vars = np.diag(cov_matrix_2)

print("Difference between variance values and eigenvalues:",(vars - eigvals).
 ↪mean())
# plt.heatmap
# np.cov(kl_transformed)
print("The value is practically zero. It is only non-zero due to floating point␣
 ↪rounding errors")
```

```
Covariance matrix:
 [[ 9.46689683e-02  3.43512262e-17 -2.67166165e-17]
 [ 3.43512262e-17  2.74486171e-02  5.97125865e-17]
 [-2.67166165e-17  5.97125865e-17  1.25373237e-02]]

Difference between variance values and eigenvalues: -1.185394375250818e-16
The value is practically zero. It is only non-zero due to floating point
rounding errors
```