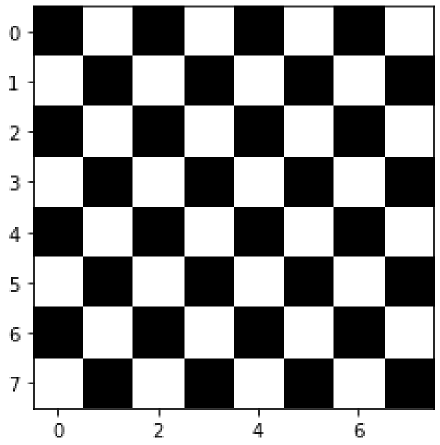
Scikit_Images

In [15]:

```
# Images are NumPy’s arrays
import numpy as np
check = np.zeros((8, 8))
check[::2, 1::2] = 2
check[1::2, ::2] = 2
import matplotlib.pyplot as plt
plt.imshow(check, cmap='gray', interpolation='nearest')
# cmap() method which returns a matplotlib color map with n colors.
# Interpolation is the process of finding a value between two points on a line or a curve
.
```

Out[15]:

```
<matplotlib.image.AxesImage at 0x1adbe053520>
```



In [6]:

```
# scikit-image is packaged in most Scientific Python distributions
```

In [13]:

```
# Python3 program to process
# images using skikit-image

# importing data from skimage
from skimage import data

camera = data.camera()

# An image with 512 rows
# and 512 columns
type(camera)

print(camera.shape)
print(camera.dtype)
```

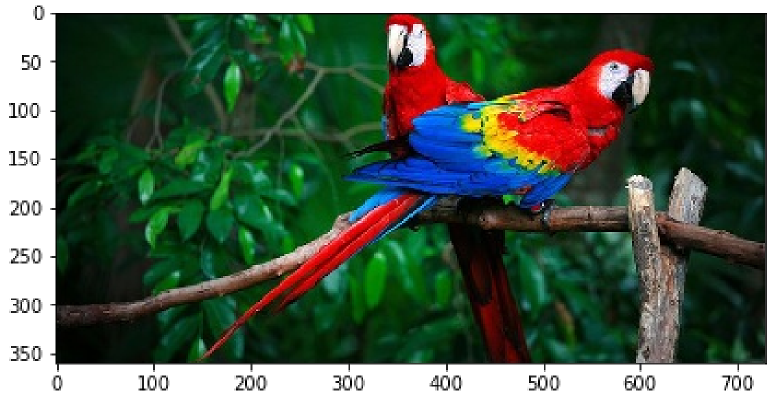
```
(512, 512)
uint8
```

In [18]:

```
# Uploading and Viewing an Image
from skimage import io
img = io.imread('parrots.jpg')
io.imshow(img)
```

Out[18]:

```
<matplotlib.image.AxesImage at 0x236288acca0>
```



In [21]:

```python
#Getting Image Resolution
from skimage import io
img = io.imread('parrots.jpg')
img.shape
```

Out[21]:

```
(360, 728, 3)
```

In [16]:

```python
#Getting Pixel Values
from skimage import io
import pandas as pd
#Pandas is used to read, write, and process various file formats.
img = io.imread('parrots.jpg')

df = pd.DataFrame(img.flatten())
# flatten function is used to convert the three dimensions of an RGB image to a single di
mension
filepath = 'pixel_values1.xlsx'
#  DataFrame function converts a one-dimensional array into an Excel-like format, with ro
ws and columns
df.to_excel(filepath, index=False)
#  to_excel save that image in an excel file.
print(df)
#print(filepath)
```

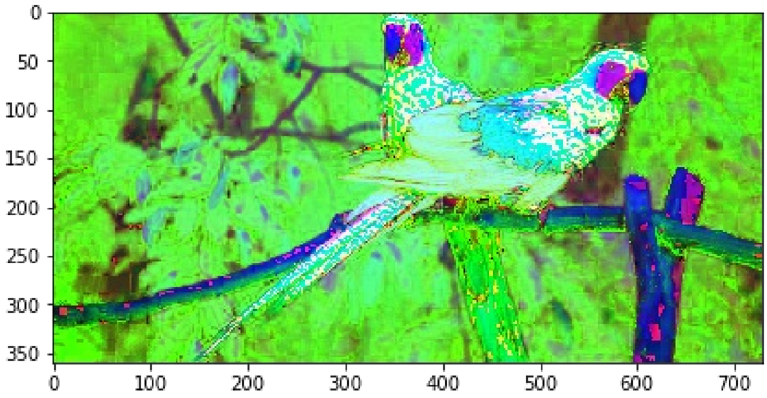
```
         0
0        5
1       18
2        0
3        6
4       19
...     ..
786235  60
786236  49
786237  25
786238  57
786239  46

[786240 rows x 1 columns]
pixel_values1.xlsx
```

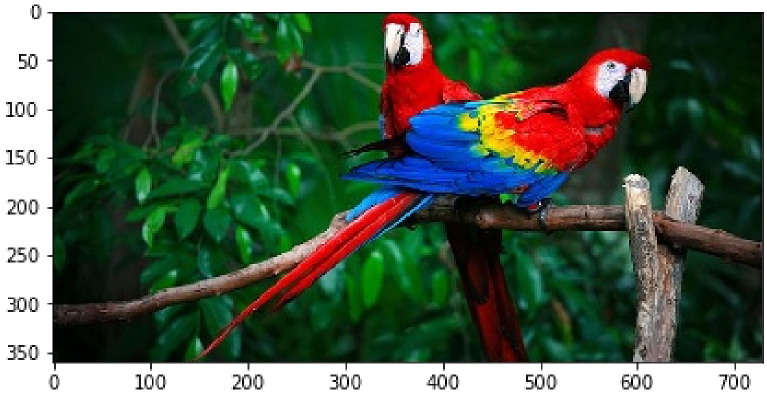
In [28]:

```python
# Converting Color Space
# pylab is used to see different figures in different blocks.

# RGB to HSV and Vice Versa

#Import libraries
from skimage import io
from skimage import color
from skimage import data
from pylab import *
#Read image
img = io.imread('parrots.jpg')
#Convert to HSV
img_hsv = color.rgb2hsv(img)
#Convert back to RGB
img_rgb = color.hsv2rgb(img_hsv)
#Show both figures

figure(0)
io.imshow(img_hsv)

figure(1)
io.imshow(img_rgb)
```

Out[28]:

```
<matplotlib.image.AxesImage at 0x23631d2c070>
```





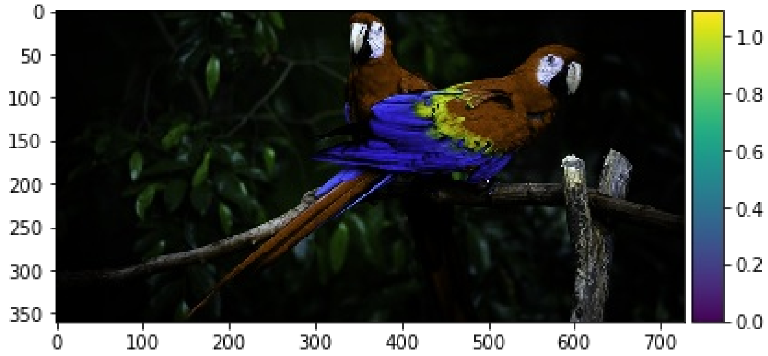
In [31]:

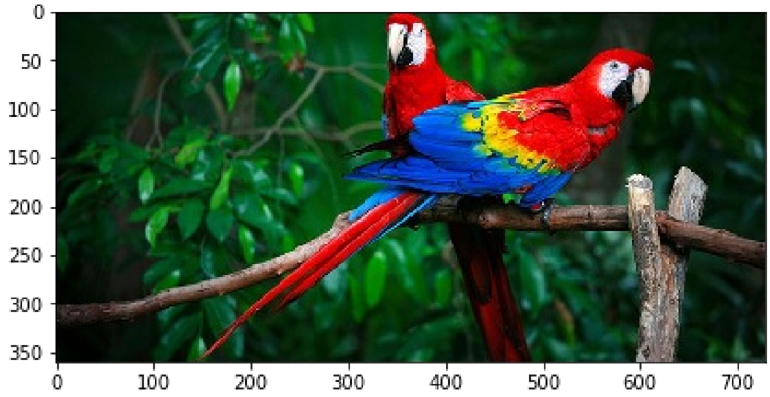
```
#RGB to XYZ and Vice Versa
#Import libraries
from skimage import io
from skimage import color
from skimage import data
#Read image
img = io.imread('parrots.jpg')
#Convert to XYZ
img_xyz = color.rgb2xyz(img)
#Convert back to RGB
img_rgb = color.xyz2rgb(img_xyz)
#Show both figures
figure(0)
io.imshow(img_xyz)
figure(1)
io.imshow(img_rgb)
```

```
C:\anaconda\lib\site-packages\skimage\io\_plugins\matplotlib_plugin.py:150: UserWarning:
Float image out of standard range; displaying image with stretched contrast.
  lo, hi, cmap = _get_display_range(image)
Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0.
.255] for integers).
```

Out[31]:

```
<matplotlib.image.AxesImage at 0x236363f4d90>
```





In [33]:

```
# Saving an Image

#Import libraries
from skimage import io
from skimage import color
from pylab import *
#Read image
img = io.imread('parrots.jpg')
#Convert to YPbPr
img_ypbpr= color.rgb2ypbpr(img)
#Convert back to RGB
img_rgb= color.ypbpr2rgb(img_ypbpr)
io.imsave("parrots_ypbpr.jpg", img_ypbpr)
```

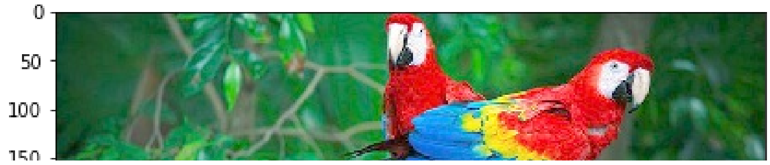
```
Lossy conversion from float64 to uint8. Range [-0.49411764705882355, 0.9991058823529411].
Convert image to uint8 prior to saving to suppress this warning.
```

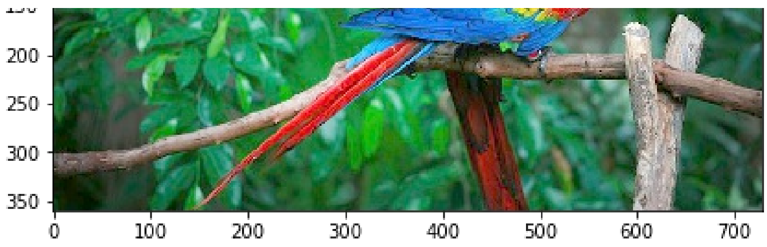
In [19]:

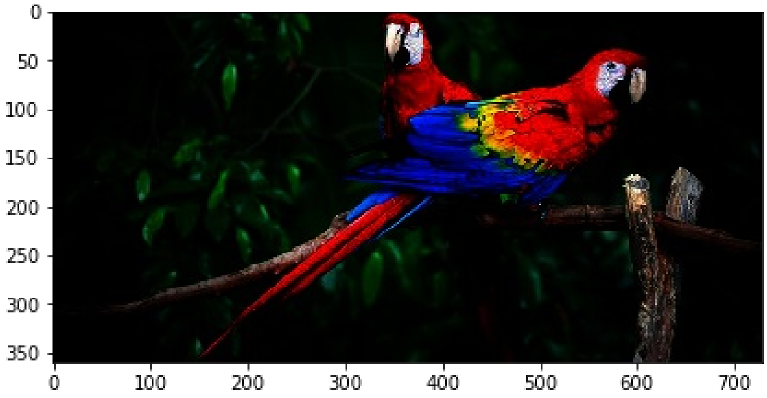
```
from skimage import exposure
from skimage import io
from pylab import *
img = io.imread('parrots.jpg')
gamma_corrected1 = exposure.adjust_gamma(img, 0.5)
gamma_corrected2 = exposure.adjust_gamma(img, 3)
figure(0)
io.imshow(gamma_corrected1)
figure(1)
io.imshow(gamma_corrected2)
```

Out[19]:

```
<matplotlib.image.AxesImage at 0x1adcb0d94f0>
```





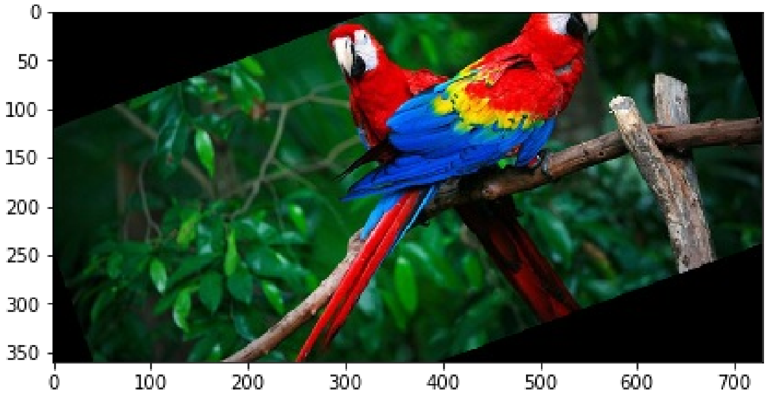


In [41]:

```
# Rotating, Shifting, and Scaling Images
from skimage import io
from skimage.transform import rotate
img = io.imread('parrots.jpg')
img_rot = rotate(img, 20)
io.imshow(img_rot)
```

Out[41]:

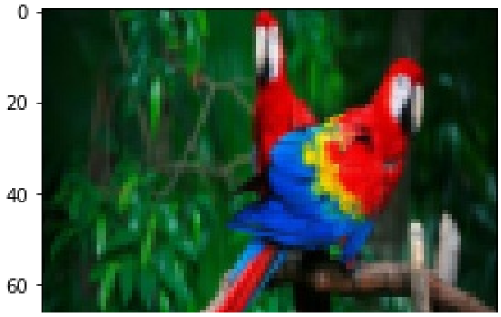
```
<matplotlib.image.AxesImage at 0x2362db91d00>
```

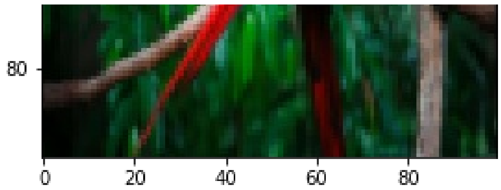


In [42]:

```
from skimage import io
from skimage.transform import resize
img = io.imread('parrots.jpg')
img_res = resize(img, (100,100))
io.imshow(img_res)
io.imsave("pp.jpg", img_res)
```

```
Lossy conversion from float64 to uint8. Range [0, 1]. Convert image to uint8 prior to sav
ing to suppress this warning.
```





In [ ]: