2023-05-08_Example-1-ColorSpaces

May 8, 2023

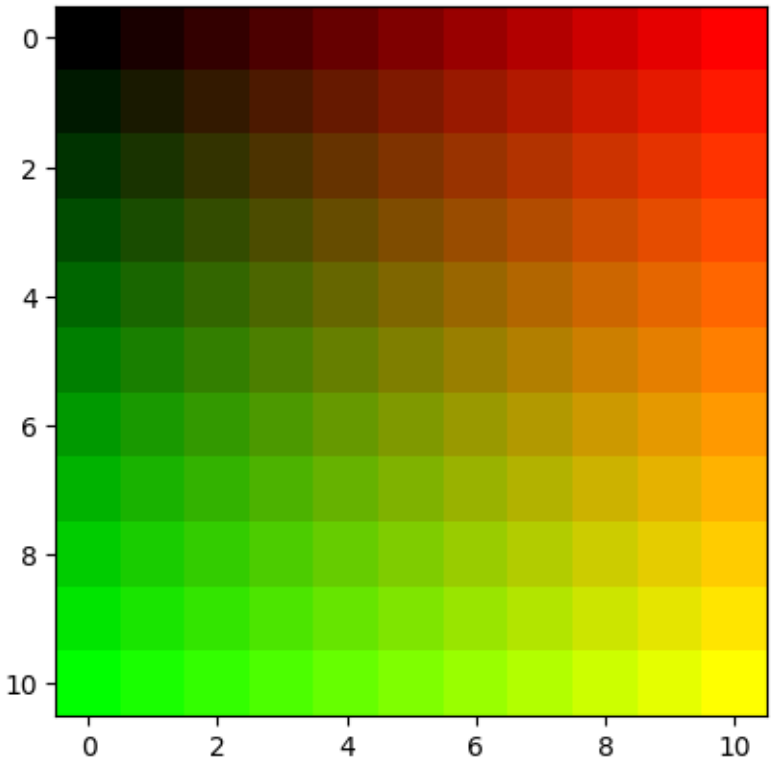
[1]:
```
import numpy as np
import scipy
import imageio

import matplotlib
import matplotlib.pyplot as plt
import matplotlib.cm as cm

matplotlib.rc('image', interpolation='nearest')
matplotlib.rc('figure',facecolor='white')
matplotlib.rc('image',cmap='viridis')

prop_cycle = plt.rcParams['axes.prop_cycle']
colors = prop_cycle.by_key()['color']

%matplotlib inline
```

1 RGB color space

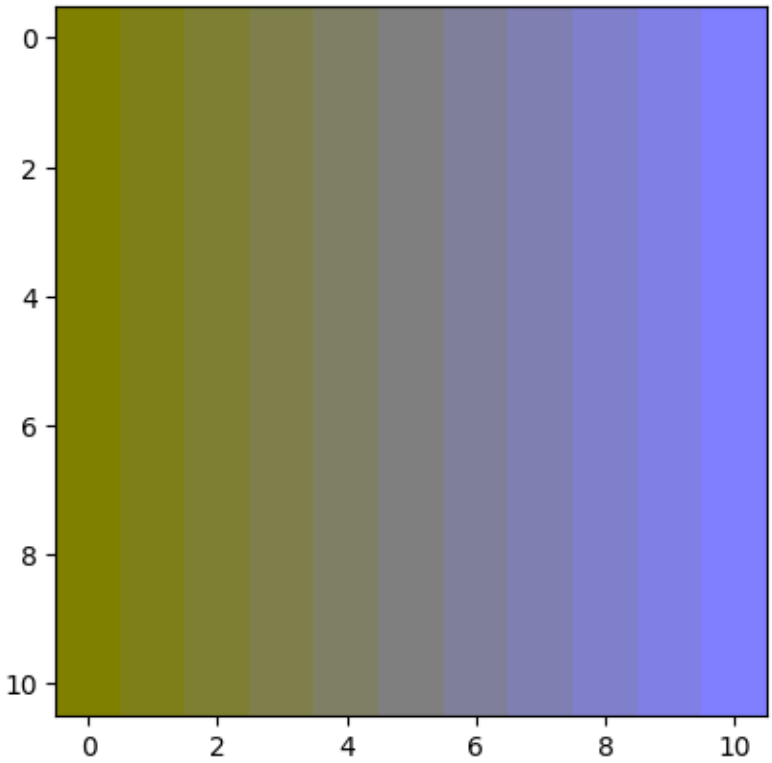
1.1 Example: simple color gradients

[2]:
```
n=11
img=np.zeros((n,n,3),dtype=np.double)
img[:,:,0]=(np.arange(n)/(n-1)).reshape((1,n))
img[:,:,1]=(np.arange(n)/(n-1)).reshape((n,1))

for i in range(n):
    for j in range(n):
        print("{:.1f}  ".format(img[i,j,1]),end="")
    print("")


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

```
0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0
0.1  0.1  0.1  0.1  0.1  0.1  0.1  0.1  0.1  0.1  0.1
0.2  0.2  0.2  0.2  0.2  0.2  0.2  0.2  0.2  0.2  0.2
0.3  0.3  0.3  0.3  0.3  0.3  0.3  0.3  0.3  0.3  0.3
0.4  0.4  0.4  0.4  0.4  0.4  0.4  0.4  0.4  0.4  0.4
0.5  0.5  0.5  0.5  0.5  0.5  0.5  0.5  0.5  0.5  0.5
0.6  0.6  0.6  0.6  0.6  0.6  0.6  0.6  0.6  0.6  0.6
0.7  0.7  0.7  0.7  0.7  0.7  0.7  0.7  0.7  0.7  0.7
0.8  0.8  0.8  0.8  0.8  0.8  0.8  0.8  0.8  0.8  0.8
0.9  0.9  0.9  0.9  0.9  0.9  0.9  0.9  0.9  0.9  0.9
1.0  1.0  1.0  1.0  1.0  1.0  1.0  1.0  1.0  1.0  1.0
```



[3]:
```
n=11
img=np.zeros((n,n,3),dtype=np.double)
img[:,:,:2]=0.5
img[:,:,2]=(np.arange(n)/(n-1)).reshape((1,n))

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



1.2 Example: load an image

[4]:
```
# image source: https://en.wikipedia.org/wiki/File:
  ↪BlueAndYellowMacaw_AraAraraunа.jpg
img=imageio.v2.imread("data/BlueAndYellowMacaw_AraAraraunа.jpg")
```

[5]:
```
img.shape
```

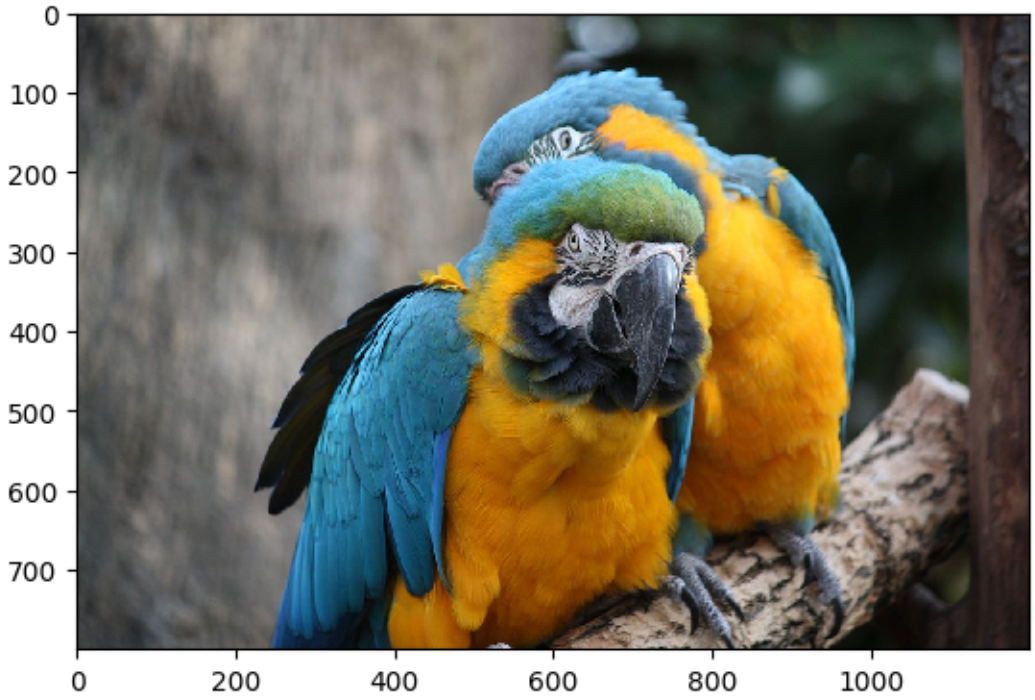
[5]: (800, 1200, 3)

[6]:
```
img.dtype
```

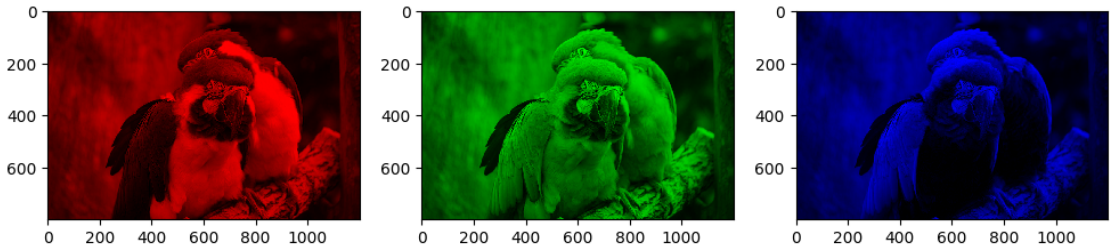
[6]: dtype('uint8')

[7]:
```
# transform to float in 0,1:
img=img/255.
```

[8]:
```
%matplotlib inline
plt.imshow(img)
plt.show()
```



[37]:
```
%matplotlib inline
# display individual RGB channels
fig=plt.figure(figsize=(12,4))
fig.add_subplot(1,3,1)
imgTest=img.copy()
imgTest[:,:,1:]=0
plt.imshow(imgTest)
fig.add_subplot(1,3,2)
imgTest=img.copy()
imgTest[:,:,0]=0
imgTest[:,:,2]=0
plt.imshow(imgTest)
fig.add_subplot(1,3,3)
imgTest=img.copy()
imgTest[:,:,:2]=0
plt.imshow(imgTest)
plt.show()
```



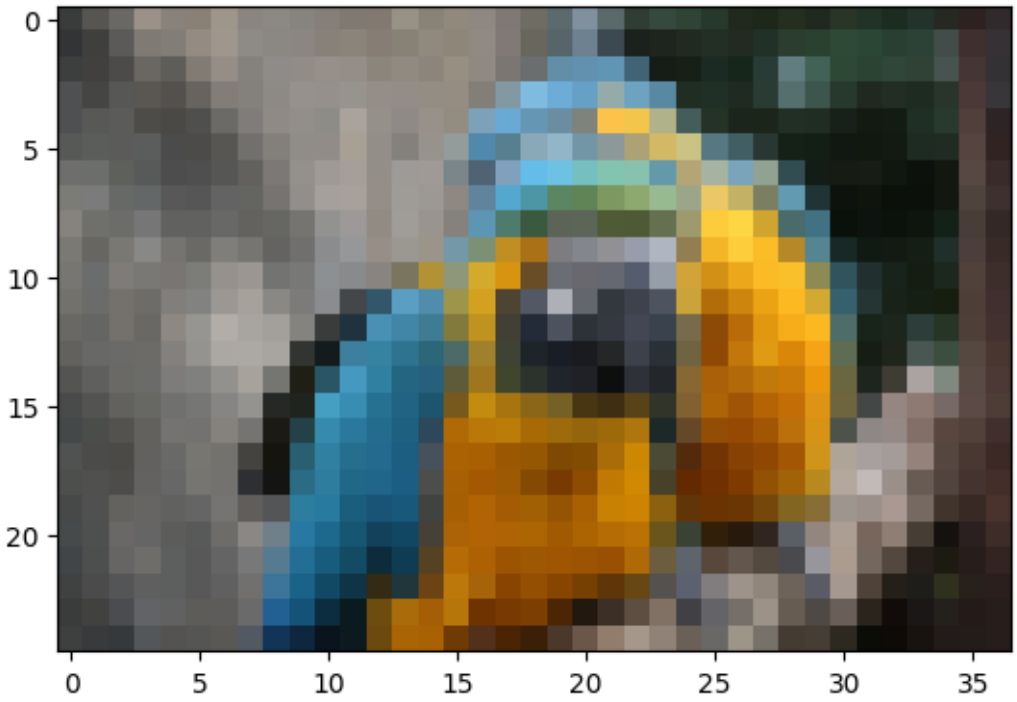
Extract color distribution

[9]:
```
# subsample image to reduce complexity a little
def subsample(dat,n):
    shape=dat.shape[:2]
    newShape=[x//n for x in shape]
    keepShape=[(x//n)*n for x in shape]
    newDat=dat[:keepShape[0],:keepShape[1]]
    interShape=(newShape[0],n,newShape[1],n)
    if dat.ndim==3:
        interShape=interShape+(dat.shape[2],)
    newDat=newDat.reshape(interShape)
    newDat=np.sum(newDat,axis=(1,3))/(n**2)
    return newDat

imgSmall=subsample(img,32)
```

[39]:
```
imgSmall.shape
```

[39]:
```
(25, 37, 3)
```

[40]:
```
plt.imshow(imgSmall)
plt.show()
```

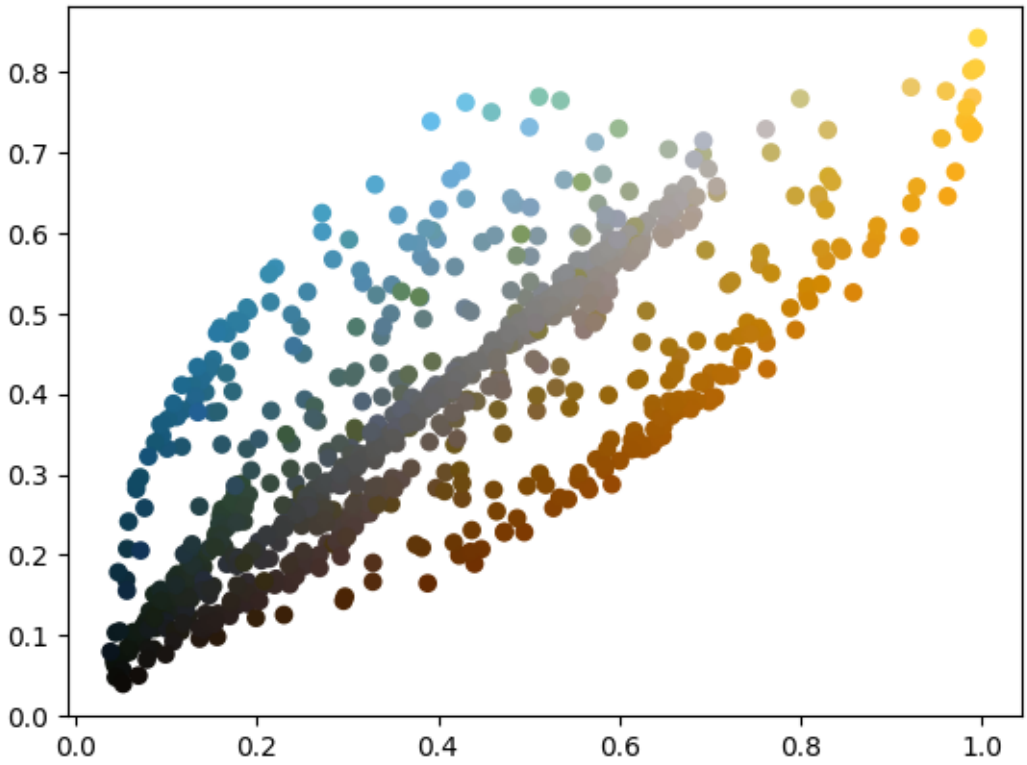


[43]:
```
pts=imgSmall.reshape((-1,3))
```

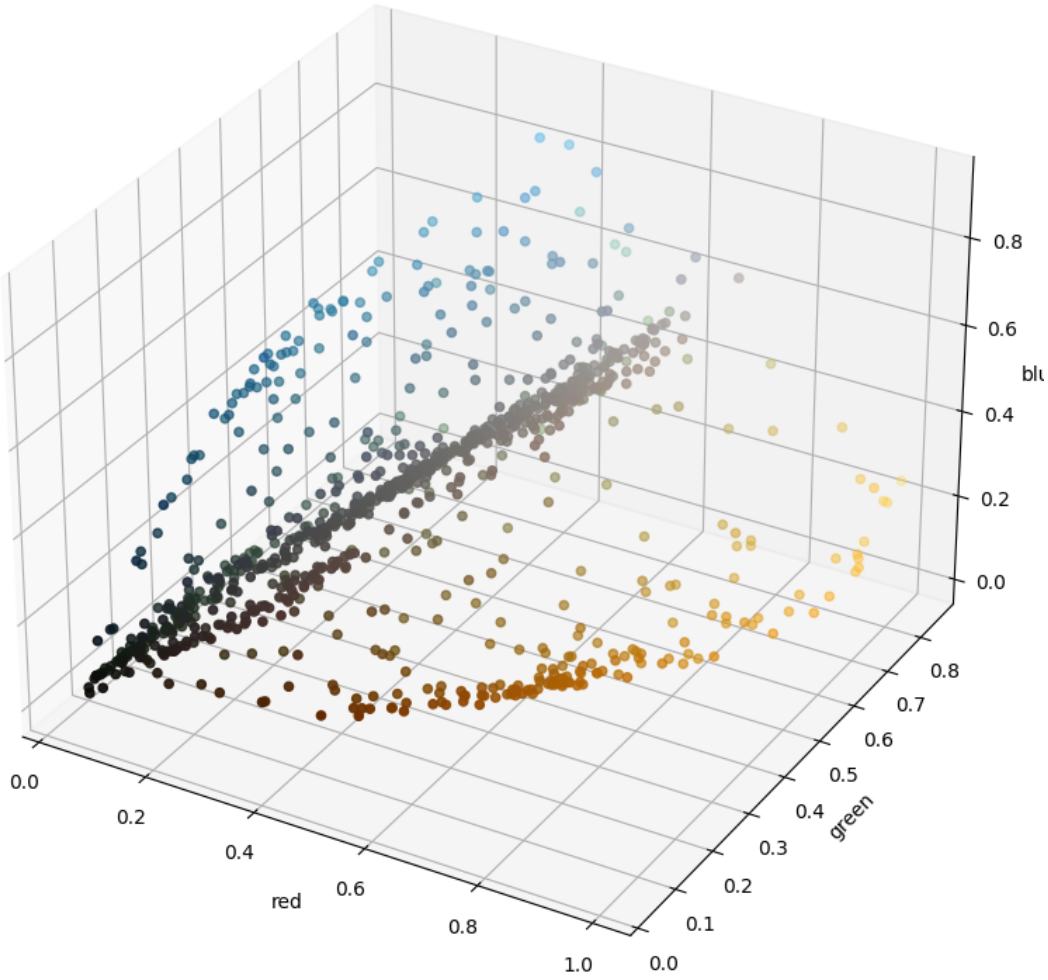
[44]:
```
pts.shape
```

[44]:
```
(925, 3)
```

[46]:
```
%matplotlib inline
plt.scatter(pts[:,0],pts[:,1],c=pts)
plt.show()
```



[47]:
```
%matplotlib widget
fig = plt.figure(figsize=(8,8))
ax = fig.add_subplot(111, projection='3d')

ax.scatter(pts[:,0],pts[:,1],pts[:,2],c=pts)
ax.set_xlabel("red")
ax.set_ylabel("green")
ax.set_zlabel("blue")
plt.tight_layout()
plt.show()
```



[48]:
```
plt.close()
%matplotlib inline
```

2 HSV

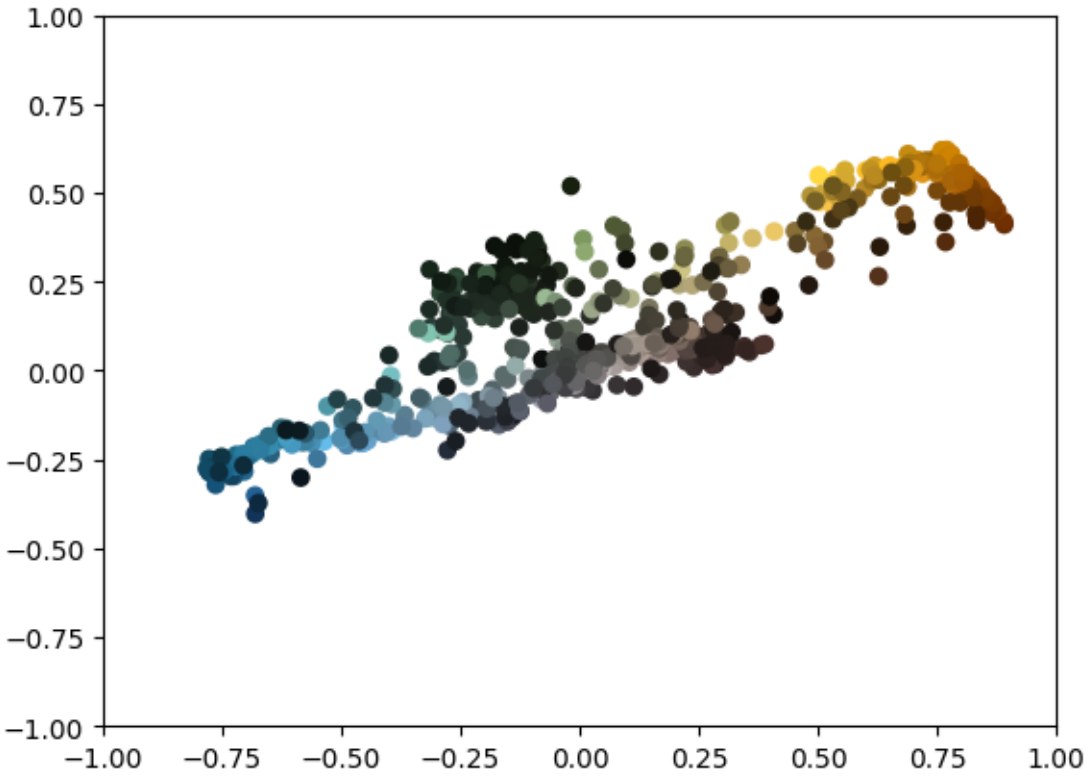
2.1 Load image and show HSV histogram

[10]:
```
# image source: https://en.wikipedia.org/wiki/File:
  ↪BlueAndYellowMacaw_AraAraraunа.jpg
img=imageio.v2.imread("data/BlueAndYellowMacaw_AraAraraunа.jpg")
img=img/255.
```

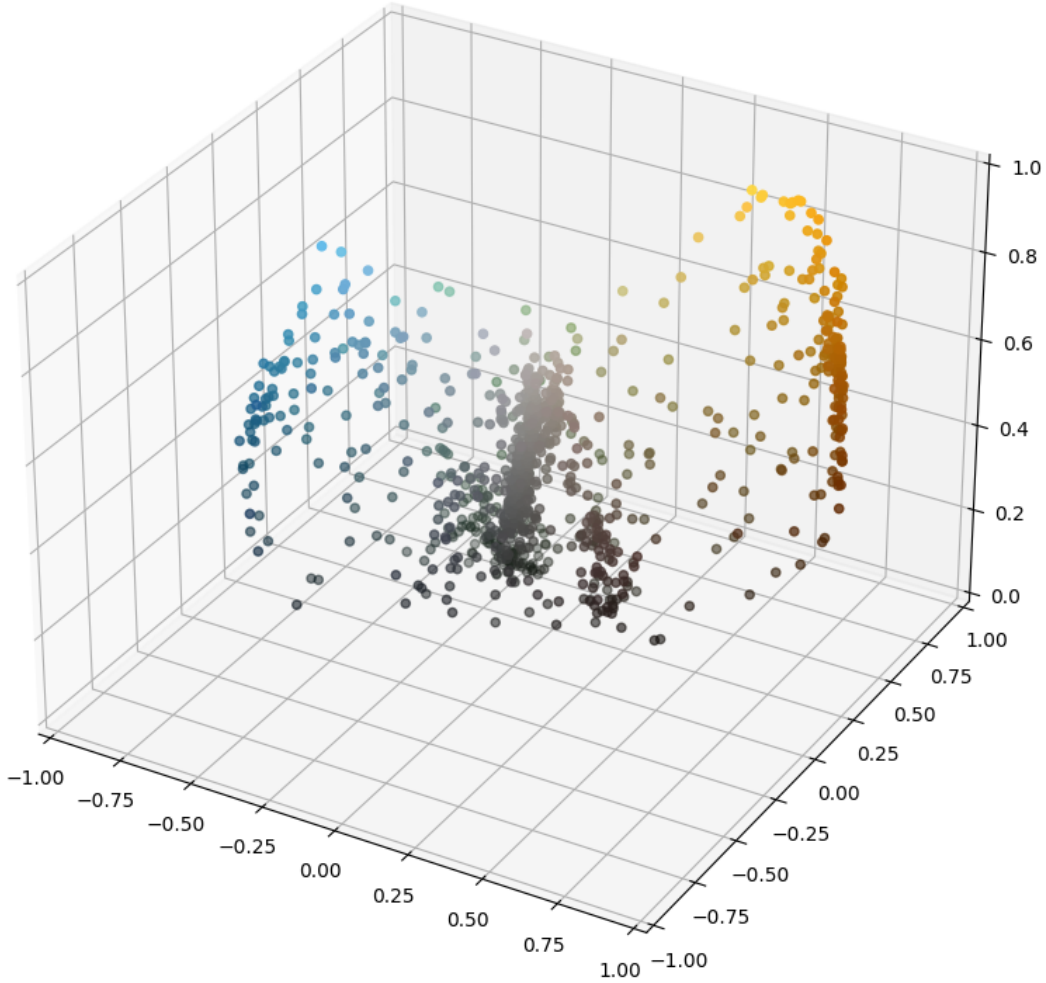
```
imgSmall=subsample(img,32)

imgHSV=matplotlib.colors.rgb_to_hsv(img)
imgSmallHSV=matplotlib.colors.rgb_to_hsv(imgSmall)
imgSmallFlat=imgSmall.reshape((-1,3))
```

[11]:
```
xres,yres=imgSmall.shape[:2]
hsvXYZ=np.zeros((xres,yres,3),dtype=np.double)
hsvXYZ[:,:,0]=imgSmallHSV[:,:,1]*np.cos(2*np.pi*imgSmallHSV[:,:,0])
hsvXYZ[:,:,1]=imgSmallHSV[:,:,1]*np.sin(2*np.pi*imgSmallHSV[:,:,0])
hsvXYZ[:,:,2]=imgSmallHSV[:,:,2]

hsvXYZ=hsvXYZ.reshape((-1,3))
```

[6]:
```
%matplotlib inline
plt.scatter(hsvXYZ[:,0],hsvXYZ[:,1],c=imgSmallFlat)
plt.xlim([-1,1])
plt.ylim([-1,1])
plt.show()
```



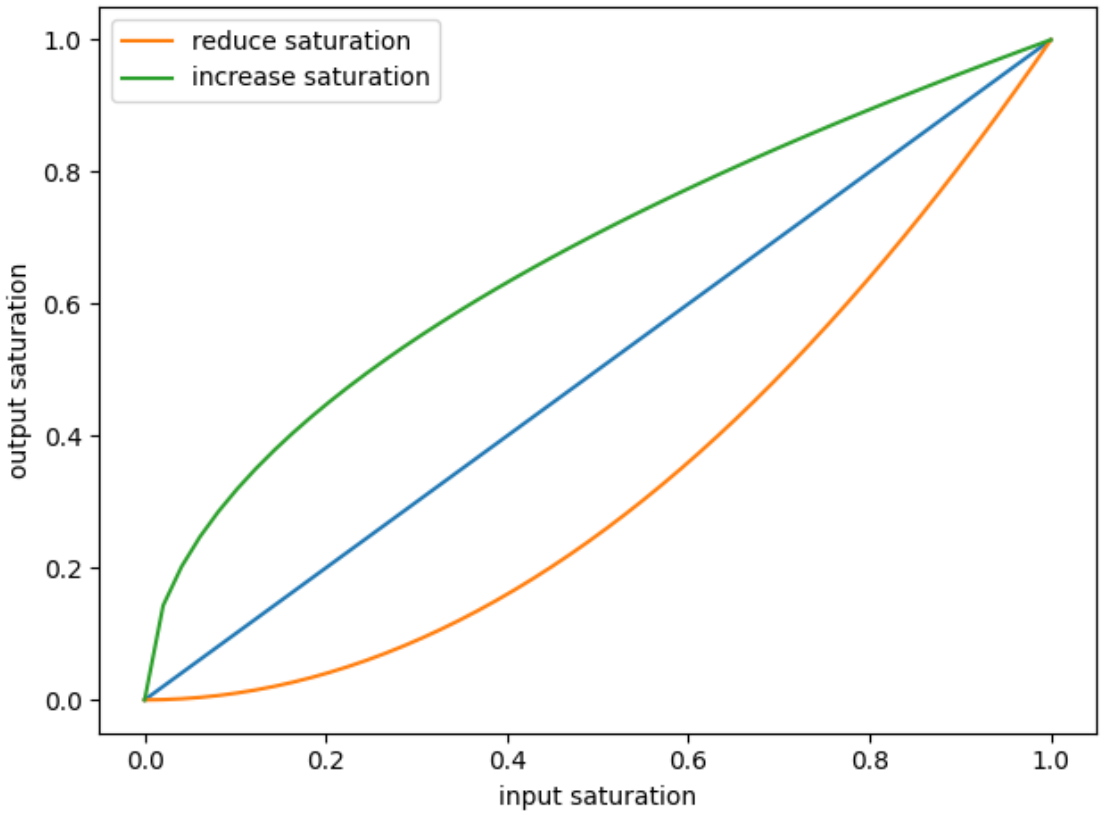
[7]:
```
%matplotlib widget
fig = plt.figure(figsize=(8,8))
ax = fig.add_subplot(111, projection='3d')

ax.scatter(hsvXYZ[:,0],hsvXYZ[:,1],hsvXYZ[:,2],c=imgSmallFlat)
ax.set_xlim([-1,1])
ax.set_ylim([-1,1])
ax.set_zlim([0,1])

plt.tight_layout()
plt.show()
```



[53]:
```
plt.close()
%matplotlib inline
```

2.2 Modify saturation

[12]:
```
%matplotlib inline
xdat=np.linspace(0,1)
plt.plot(xdat,xdat)
plt.plot(xdat,xdat**2,label="reduce saturation")
plt.plot(xdat,xdat**0.5,label="increase saturation")
plt.xlabel("input saturation")
plt.ylabel("output saturation")
plt.legend()
plt.tight_layout()
plt.show()
```



[13]:
```
# image source: https://en.wikipedia.org/wiki/File:
 ↪BlueAndYellowMacaw_AraAraraunа.jpg
img=imageio.v2.imread("data/BlueAndYellowMacaw_AraAraraunа.jpg")
```

```
img=img/255.

imgHSV=matplotlib.colors.rgb_to_hsv(img)
imgHSV1=imgHSV.copy()
imgHSV2=imgHSV.copy()
imgHSV3=imgHSV.copy()

# reduce saturation
imgHSV1[:,:,1]=imgHSV1[:,:,1]**2
# increase saturation
imgHSV2[:,:,1]=imgHSV2[:,:,1]**.5

# set color to pure hue
imgHSV3[:,:,1]=1.
imgHSV3[:,:,2]=1.

img1=matplotlib.colors.hsv_to_rgb(imgHSV1)
img2=matplotlib.colors.hsv_to_rgb(imgHSV2)
img3=matplotlib.colors.hsv_to_rgb(imgHSV3)
```

[14]:
```
%matplotlib inline
# display individual RGB channels
fig=plt.figure(figsize=(16,4))
fig.add_subplot(1,4,1)
plt.imshow(img)
plt.axis("off")
fig.add_subplot(1,4,2)
plt.imshow(img1)
plt.axis("off")
fig.add_subplot(1,4,3)
plt.imshow(img2)
plt.axis("off")
fig.add_subplot(1,4,4)
plt.imshow(img3)
plt.axis("off")
plt.tight_layout()
plt.show()
```

[ ]: