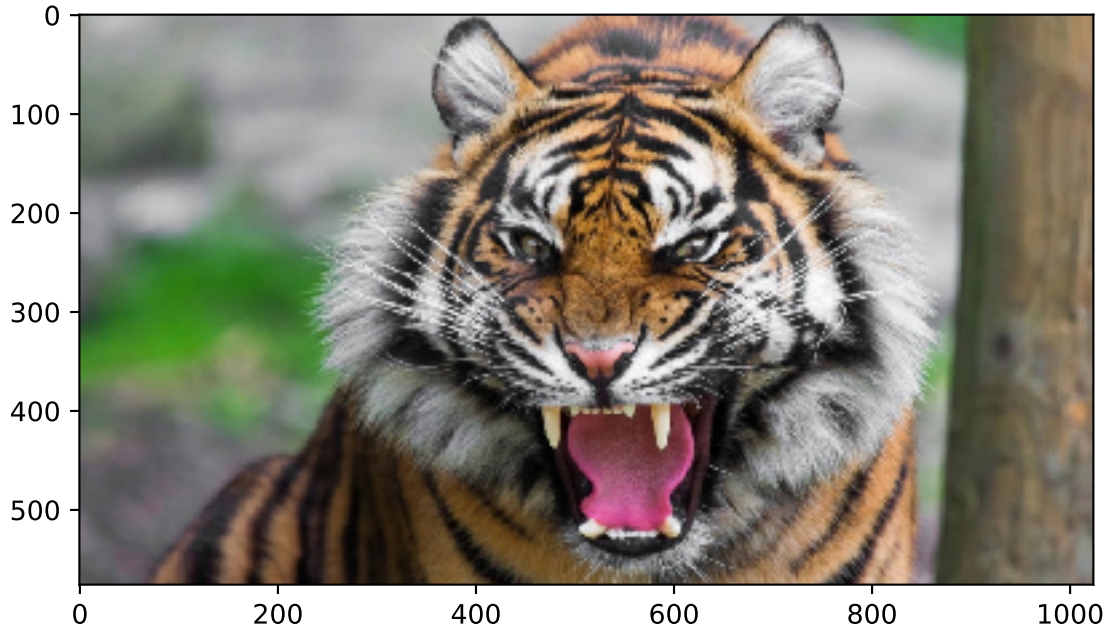
In [17]:
```
#Load image
import numpy as np
from skimage import io
tiger = io.imread("tiger.png")
io.imshow(tiger)
io.show()
print(tiger.shape)
a,b,c = tiger.shape

#reshaping image array from 3D to 2D for k-mean clustering algo
tiger = np.reshape(tiger,(a * b,c))
print(tiger.shape)
print(tiger)
```




```
(576, 1024, 3)
(589824, 3)
[[164 160 159]
 [164 162 160]
 [165 164 162]
 ...
 [119 115 106]
 [120 116 107]
 [122 118 109]]
```

In [13]:
```
# Fitting Model on loaded image
from sklearn.cluster import KMeans
kmeans = KMeans(n_clusters=64,random_state=0).fit(tiger)
print(kmeans.cluster_centers_)
print(kmeans.labels_)
```

```
[[203.97390406 204.27051934 205.44208079]
 [129.21311475  93.56404251  62.05602594]
 [128.79108705 132.72986256 130.21524365]
 [ 62.98718613  65.08117917  48.5630694 ]
 [174.95050271 176.1920437  177.11146558]
 [ 23.76133721  18.11155523  15.94949128]
 [119.45312147 103.63935537  81.89886287]
 [ 64.59597599 124.92081778  54.18059387]
 [156.97663874 160.16205027 159.27821587]
 [ 89.71843275  90.99308566  88.83273885]
 [166.08257865 132.56160629 101.90228789]
```

```
 [209.86934477 158.30947738 112.42765211]
 [ 89.42937269  83.95874539  67.91261993]
 [117.86760072 124.45163237 117.70067708]
 [226.57536467 225.57316508 225.91815235]
 [177.74524715  81.43060837 121.12642586]
 [100.03374814 101.49213909  98.23511036]
 [ 53.81815447  50.25073325  49.92735203]
 [218.27194492 172.23541786 133.69152802]
 [ 95.37062647  66.45775135  34.91605587]
 [148.85215139 152.28369807 151.03118723]
 [134.79104922 112.90341798  85.92277906]
 [ 98.68153637 133.87887494  99.08619386]
 [167.60960384 119.94645858  70.74045618]
 [239.76270897 239.61975435 240.49351757]
 [205.3192046   92.84929356 149.27995814]
 [194.01784192 194.41311906 195.45149229]
 [ 68.79203335 143.21120889  33.14034275]
 [119.73427673  60.23506289  69.72602201]
 [ 43.06098964  28.21944764  17.47238205]
 [224.55587229 202.92474344 184.72833523]
 [ 10.7963574    8.41018873   7.88214333]
 [189.60421422 159.37278314 137.77524144]
 [ 49.3481153  122.04617191  31.56110604]
 [201.72883117 179.22311688 160.67506494]
 [105.26464496  92.15096528  74.58223548]
 [166.36572062 168.13455672 168.42738282]
 [ 58.95657603  40.55133185  28.41745283]
 [121.11660921 146.31170148  94.31819104]
 [145.71388889  69.96574074  95.45416667]
 [185.71518882 145.94942605 115.73332224]
 [150.90499351 101.10343709  47.21108949]
 [ 66.02166133  63.56601997  63.90930495]
 [ 50.45246753 104.51454545  42.6025974 ]
 [184.21795455 184.95931818 185.84153409]
 [139.02498898 143.69373806 140.97640747]
 [ 30.86979259  27.43822017  28.19393725]
 [ 77.68463143  78.44567107  76.88577886]
 [113.6786881   80.19195087  43.73435254]
 [ 78.01562961  51.90209378  29.93482748]
 [230.02180867 190.05524862 154.38383251]
 [192.62160778 141.64311316  88.375064  ]
 [ 77.86029256 131.27220691  76.33008268]
 [ 77.00158311  75.07749717  57.51903505]
 [ 99.29540851  47.40850777  47.12424038]
 [109.68384393 110.16371818 109.79196203]
 [145.53395034 122.14051793  95.74708552]
 [135.32371885 124.24026883 112.52590311]
 [214.87142281 214.00806127 214.27559452]
 [122.37690447 113.30592503  99.30931076]
 [129.82535684 153.11125105 119.1702351 ]
 [ 99.24447101 144.45517035  68.61237298]
 [159.14478976 136.9226691  124.43528336]
 [ 41.58448433  38.61186006  39.03382681]]
[ 8  8 36 ... 59 59 59]
```

In [14]:

```
#saving compressed image and codebook(like a color pallete)
cluster_centroid = np.array(kmeans.cluster_centers_,dtype=np.uint8)
print(cluster_centroid)
labels = np.array(kmeans.labels_,dtype=np.uint8)
labels = np.reshape(labels,(a, b))
print(labels)
np.save("codebook.npy",cluster_centroid)
io.imsave("compressed_tiger.png",labels)
```

```
[[203 204 205]
 [129  93  62]
 [128 132 130]
 [ 62  65  48]
 [174 176 177]
 [ 23  18  15]
 [119 103  81]
 [ 64 124  54]
 [156 160 159]
 [ 89  90  88]
 [166 132 101]
 [209 158 112]
 [ 89  83  67]
 [117 124 117]
 [226 225 225]
 [177  81 121]
 [100 101  98]
 [ 53  50  49]
 [218 172 133]
 [ 95  66  34]
 [148 152 151]
 [134 112  85]
 [ 98 133  99]
 [167 119  70]
 [239 239 240]
 [205  92 149]
 [194 194 195]
 [ 68 143  33]
 [119  60  69]
 [ 43  28  17]
 [224 202 184]
 [ 10   8   7]
 [189 159 137]
 [ 49 122  31]
 [201 179 160]
 [105  92  74]
 [166 168 168]
 [ 58  40  28]
 [121 146  94]
 [145  69  95]
 [185 145 115]
 [150 101  47]
 [ 66  63  63]
 [ 50 104  42]
 [184 184 185]
 [139 143 140]
 [ 30  27  28]
 [ 77  78  76]
 [113  80  43]
 [ 78  51  29]
 [230 190 154]
 [192 141  88]
 [ 77 131  76]
 [ 77  75  57]
 [ 99  47  47]
 [109 110 109]
 [145 122  95]
 [135 124 112]
 [214 214 214]
 [122 113  99]
 [129 153 119]
 [ 99 144  68]
 [159 136 124]
 [ 41  38  39]]]
[[ 8  8 36 ... 10 10 10]
```

```
[ 8  8  8 ... 10 10 56]
[ 8  8  8 ... 10 10 10]
...
[ 9  9  9 ... 59 59 59]
[ 9  9  9 ... 59 59 59]
[ 9  9  9 ... 59 59 59]]
```

In [16]:

```
#decompress the compressed image
compressed_tiger = io.imread("compressed_tiger.png")
codebook = np.load("codebook.npy")

decompressed_image = np.zeros((a,b,3),dtype= np.uint8)

for i in range(a):
    for j in range(b):
        decompressed_image[i,j,:] = codebook[compressed_tiger[i,j],:]

io.imshow(decompressed_image)
io.show()
io.imsave("decompressed_image.png",decompressed_image)

# now decompress image size on disk is also less than original image
```