In [24]:

```
import os
import warnings
warnings.simplefilter('ignore')
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
from skimage.io import imread, imshow
from skimage.transform import resize
from skimage.color import rgb2gray
```

In [25]:

```
ch=os.listdir("C:/Users/SRIVENKATESH/Downloads/CEC/chair")
ca=os.listdir("C:/Users/SRIVENKATESH/Downloads/CEC/car")
el=os.listdir("C:/Users/SRIVENKATESH/Downloads/CEC/elephant")
```

In [26]:

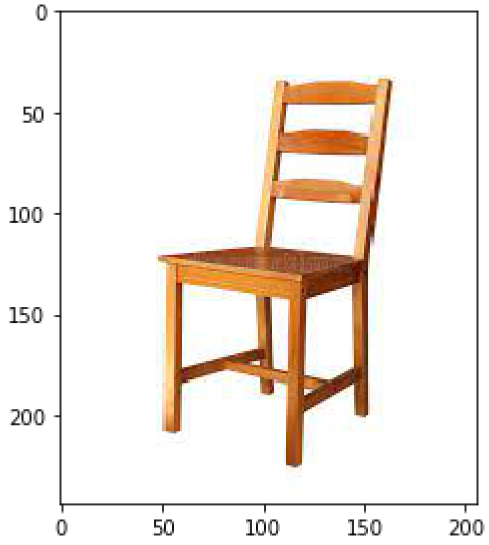
```
limit=10
ch_image=[None]*limit
j=0
for i in ch:
    if(j<limit):
        ch_image[j]=imread("C:/Users/SRIVENKATESH/Downloads/CEC/chair/"+i)
        j+=1
    else:
        break
limit=10
ca_image=[None]*limit
j=0
for i in ca:
    if(j<limit):
        ca_image[j]=imread("C:/Users/SRIVENKATESH/Downloads/CEC/car/"+i)
        j+=1
    else:
        break
    limit=10
el_image=[None]*limit
j=0
for i in el:
    if(j<limit):
        el_image[j]=imread("C:/Users/SRIVENKATESH/Downloads/CEC/elephant/"+i)
        j+=1
    else:
        break
```

In [27]:

```
imshow(ch_image[4])
```

Out[27]:

```
<matplotlib.image.AxesImage at 0x20ac32205b0>
```



In [28]:

```
imshow(ca_image[4])
```

Out[28]:

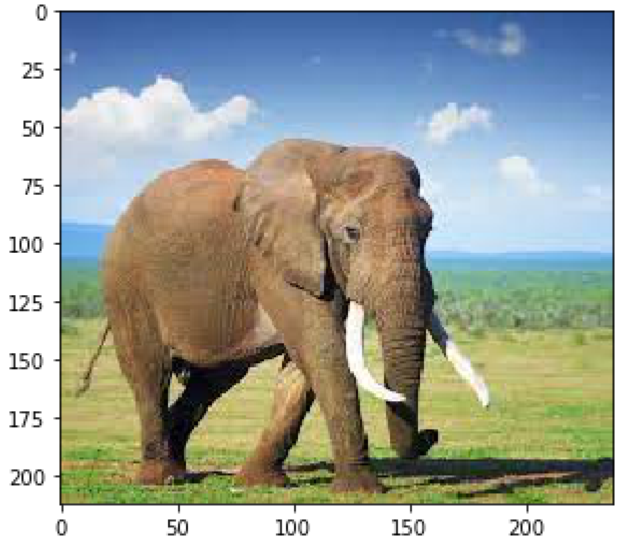
```
<matplotlib.image.AxesImage at 0x20ac35d9670>
```

In [29]:

```
imshow(el_image[4])
```

Out[29]:

```
<matplotlib.image.AxesImage at 0x20abded20d0>
```



In [30]:

```
ch_image[4].shape
```

Out[30]:

```
(244, 206, 3)
```

In [31]:

```
ca_image[4].shape
```

Out[31]:

```
(168, 300, 3)
```

In [32]:

```
el_image[4].shape
```

Out[32]:

```
(212, 238, 3)
```

In [33]:

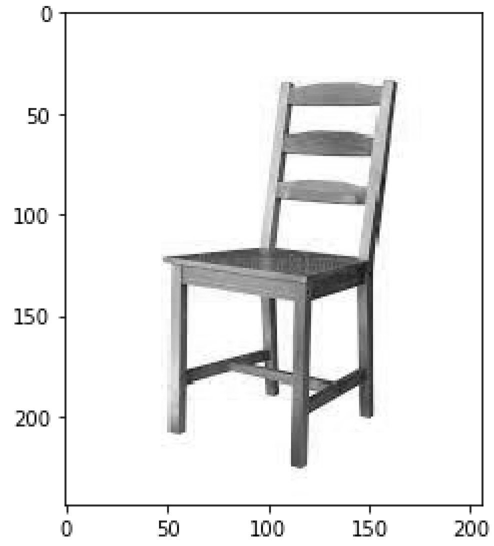
```
ch_gray=[None]*limit
j=0
for i in ch:
    if(j<limit):
        ch_gray[j]=rgb2gray(ch_image[j])
        j+=1
    else:
        break
ca_gray=[None]*limit
j=0
for i in ca:
    if(j<limit):
        ca_gray[j]=rgb2gray(ca_image[j])
        j+=1
    else:
        break
el_gray=[None]*limit
j=0
for i in el:
    if(j<limit):
        el_gray[j]=rgb2gray(el_image[j])
        j+=1
    else:
        break
```

In [34]:

```
imshow(ch_gray[4])
```

Out[34]:

```
<matplotlib.image.AxesImage at 0x20abdf33cd0>
```



In [35]:

```
imshow(ca_gray[4])
```

Out[35]:

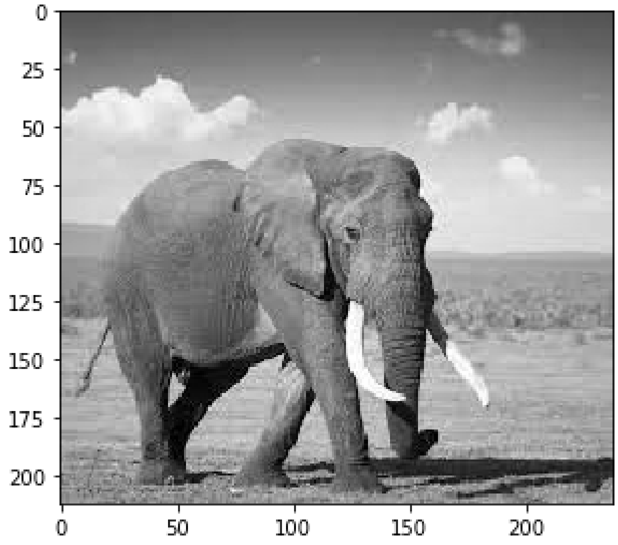
```
<matplotlib.image.AxesImage at 0x20abe0f32b0>
```

In [36]:

```
imshow(el_gray[4])
```

Out[36]:

```
<matplotlib.image.AxesImage at 0x20ac01384f0>
```



In [37]:

```
ch_gray[4].shape
```

Out[37]:

```
(244, 206)
```

In [38]:

```
ca_gray[4].shape
```

Out[38]:

```
(168, 300)
```

In [39]:

```
el_gray[4].shape
```

Out[39]:

```
(212, 238)
```

In [43]:

```
for j in range(10):
    ch=ch_gray[j]
    ch_gray[j]=resize(ch,(512,512))
```

In [44]:

```
for j in range(10):
    ca=ca_gray[j]
    ca_gray[j]=resize(ca,(512,512))
```

In [45]:

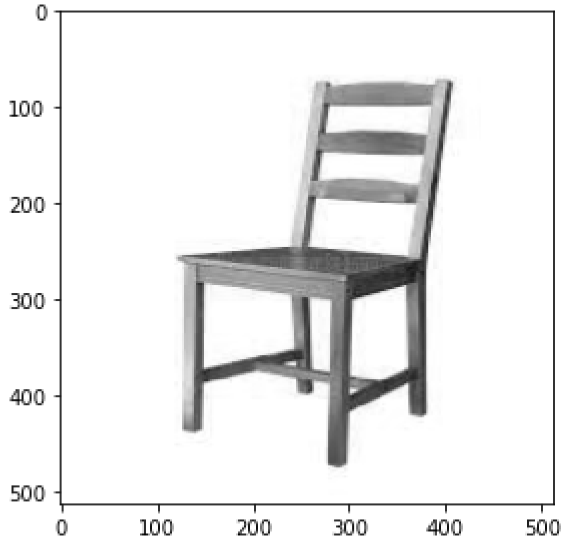
```
for j in range(10):
    el=el_gray[j]
    el_gray[j]=resize(el,(512,512))
```

In [46]:

```
imshow(ch_gray[4])
```

Out[46]:

```
<matplotlib.image.AxesImage at 0x20ac01a6130>
```

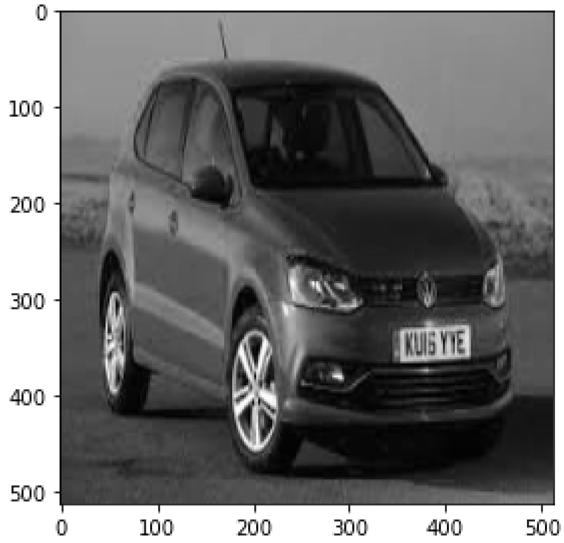


In [47]:

```
imshow(ca_gray[4])
```

Out[47]:

```
<matplotlib.image.AxesImage at 0x20abdf9a160>
```

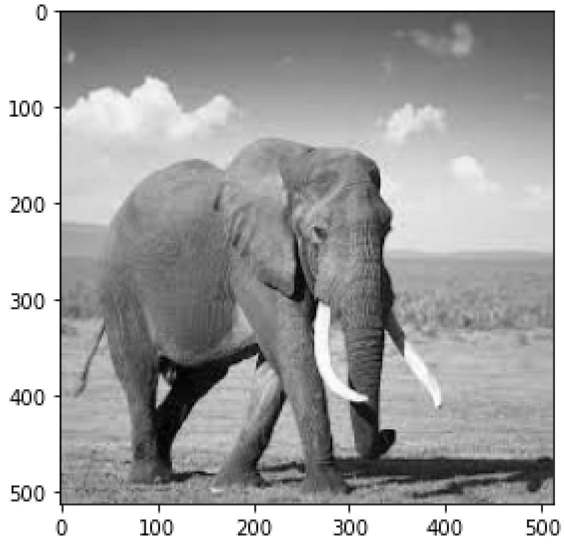


In [48]:

```
imshow(el_gray[4])
```

Out[48]:

```
<matplotlib.image.AxesImage at 0x20abdffd130>
```



In [49]:

```
len_of_image_ch=len(ch_gray)
len_of_image_ch
len_of_image_ca=len(ca_gray)
len_of_image_ca
len_of_image_el=len(el_gray)
len_of_image_el
```

Out[49]:

```
10
```

In [50]:

```
image_size_ca=ca_gray[4].shape
image_size_ca
image_size_el=el_gray[4].shape
image_size_el
image_size_ch=ch_gray[4].shape
image_size_ch
```

Out[50]:

```
(512, 512)
```

In [51]:

```
flatten_size_ca=image_size_ca[0]*image_size_ca[1]
flatten_size_ca
flatten_size_el=image_size_el[0]*image_size_el[1]
flatten_size_el
flatten_size_ch=image_size_ch[0]*image_size_ch[1]
flatten_size_ch
```

Out[51]:

```
262144
```

In [52]:

```
for i in range(len_of_image_ca):
    ca_gray[i]=np.ndarray.flatten(ca_gray[i]).reshape(flatten_size_ca,1)
ca_gray[4].shape
for i in range(len_of_image_el):
    el_gray[i]=np.ndarray.flatten(el_gray[i]).reshape(flatten_size_el,1)
el_gray[4].shape
for i in range(len_of_image_ch):
    ch_gray[i]=np.ndarray.flatten(ch_gray[i]).reshape(flatten_size_ch,1)
ch_gray[4].shape
```

Out[52]:

```
(262144, 1)
```

In [53]:

```
ca_gray=np.dstack(ca_gray)
el_gray=np.dstack(el_gray)
ch_gray=np.dstack(ch_gray)
```

In [54]:

```
ca_gray=np.rollaxis(ca_gray,axis=2,start=0)
ca_gray.shape
el_gray=np.rollaxis(el_gray,axis=2,start=0)
el_gray.shape
ch_gray=np.rollaxis(ch_gray,axis=2,start=0)
ch_gray.shape
```

Out[54]:

```
(10, 262144, 1)
```

In [56]:

```
ca_gray=ca_gray.reshape(len_of_image_ca,flatten_size_ca)
ca_gray.shape
el_gray=el_gray.reshape(len_of_image_el,flatten_size_el)
el_gray.shape
ch_gray=ch_gray.reshape(len_of_image_ch,flatten_size_ch)
ch_gray.shape
```

Out[56]:

```
(10, 262144)
```

In [57]:

```
ca_data=pd.DataFrame(ca_gray)
ca_data
```

Out[57]:

	0	1	2	3	4	5	6	7	8
0	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
1	0.988550	0.988550	0.988550	0.988550	0.988550	0.988550	0.988550	0.988550	0.988550
2	0.111776	0.111776	0.111776	0.111776	0.111776	0.111776	0.111776	0.111776	0.111776
3	0.943523	0.944489	0.951485	0.946030	0.935423	0.933267	0.932656	0.938787	0.941648
4	0.561135	0.561135	0.561135	0.563294	0.565056	0.565056	0.566266	0.568564	0.570862
5	0.630525	0.632752	0.638669	0.646816	0.652649	0.655925	0.654333	0.650120	0.640460
6	0.225986	0.245332	0.306594	0.304438	0.287543	0.238678	0.209466	0.193071	0.185980
7	0.432715	0.443749	0.374935	0.339013	0.396952	0.445157	0.404925	0.364693	0.381725
8	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
9	0.763658	0.763658	0.763658	0.763658	0.763658	0.763658	0.763658	0.763658	0.763658

10 rows × 262144 columns

In [58]:

```
el_data=pd.DataFrame(el_gray)
el_data
```

Out[58]:

	0	1	2	3	4	5	6	7	8
0	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
1	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
2	0.911239	0.910965	0.908327	0.909763	0.913341	0.920210	0.928124	0.935630	0.943071
3	0.762583	0.762791	0.764226	0.764900	0.765274	0.767148	0.769255	0.770306	0.771415
4	0.339931	0.339931	0.339931	0.339931	0.339931	0.339931	0.339931	0.339931	0.339931
5	0.811747	0.811747	0.811747	0.811747	0.811747	0.811747	0.811747	0.811747	0.811747
6	0.968123	0.967495	0.968642	0.969790	0.970424	0.970817	0.971157	0.971157	0.971157
7	0.540873	0.540873	0.540873	0.540873	0.540873	0.540873	0.540873	0.540873	0.540873
8	0.996451	0.994424	0.985269	0.980963	0.982586	0.985624	0.984999	0.979978	0.981284
9	0.421518	0.423566	0.418915	0.398364	0.373263	0.360707	0.372740	0.397506	0.399863

10 rows × 262144 columns

In [59]:

```
ch_data=pd.DataFrame(ch_gray)
ch_data
```

Out[59]:

	0	1	2	3	4	5	6	7	8
0	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
1	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
2	0.659031	0.659031	0.659031	0.659170	0.660769	0.662368	0.662670	0.662670	0.663593
3	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
4	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
5	0.949020	0.949020	0.949020	0.949020	0.949020	0.949020	0.949020	0.949020	0.949020
6	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
7	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
8	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
9	0.604165	0.603629	0.605322	0.606972	0.606972	0.606972	0.608171	0.609864	0.611556

10 rows × 262144 columns

In [60]:

```
ca_data["label"]="ca"
ca_data
```

Out[60]:

	0	1	2	3	4	5	6	7	8
0	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
1	0.988550	0.988550	0.988550	0.988550	0.988550	0.988550	0.988550	0.988550	0.988550
2	0.111776	0.111776	0.111776	0.111776	0.111776	0.111776	0.111776	0.111776	0.111776
3	0.943523	0.944489	0.951485	0.946030	0.935423	0.933267	0.932656	0.938787	0.941648
4	0.561135	0.561135	0.561135	0.563294	0.565056	0.565056	0.566266	0.568564	0.570862
5	0.630525	0.632752	0.638669	0.646816	0.652649	0.655925	0.654333	0.650120	0.640460
6	0.225986	0.245332	0.306594	0.304438	0.287543	0.238678	0.209466	0.193071	0.185980
7	0.432715	0.443749	0.374935	0.339013	0.396952	0.445157	0.404925	0.364693	0.381725
8	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
9	0.763658	0.763658	0.763658	0.763658	0.763658	0.763658	0.763658	0.763658	0.763658

10 rows × 262145 columns

In [61]:

```
el_data["label"]="el"
el_data
```

Out[61]:

	0	1	2	3	4	5	6	7	8	
0	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1
1	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1
2	0.911239	0.910965	0.908327	0.909763	0.913341	0.920210	0.928124	0.935630	0.943071	0
3	0.762583	0.762791	0.764226	0.764900	0.765274	0.767148	0.769255	0.770306	0.771415	0
4	0.339931	0.339931	0.339931	0.339931	0.339931	0.339931	0.339931	0.339931	0.339931	0
5	0.811747	0.811747	0.811747	0.811747	0.811747	0.811747	0.811747	0.811747	0.811747	0
6	0.968123	0.967495	0.968642	0.969790	0.970424	0.970817	0.971157	0.971157	0.971157	0
7	0.540873	0.540873	0.540873	0.540873	0.540873	0.540873	0.540873	0.540873	0.540873	0
8	0.996451	0.994424	0.985269	0.980963	0.982586	0.985624	0.984999	0.979978	0.981284	0
9	0.421518	0.423566	0.418915	0.398364	0.373263	0.360707	0.372740	0.397506	0.399863	0

10 rows × 262145 columns

In [62]:

```
ch_data["label"]="ch"
ch_data
```

Out[62]:

	0	1	2	3	4	5	6	7	8	
0	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1
1	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1
2	0.659031	0.659031	0.659031	0.659170	0.660769	0.662368	0.662670	0.662670	0.663593	0
3	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1
4	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1
5	0.949020	0.949020	0.949020	0.949020	0.949020	0.949020	0.949020	0.949020	0.949020	0
6	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1
7	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1
8	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1
9	0.604165	0.603629	0.605322	0.606972	0.606972	0.606972	0.608171	0.609864	0.611556	0

10 rows × 262145 columns

In [63]:

```
actor_1=pd.concat([ca_data,el_data,ch_data])
actor_1
```

Out[63]:

	0	1	2	3	4	5	6	7	8	
0	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1
1	0.988550	0.988550	0.988550	0.988550	0.988550	0.988550	0.988550	0.988550	0.988550	0
2	0.111776	0.111776	0.111776	0.111776	0.111776	0.111776	0.111776	0.111776	0.111776	0
3	0.943523	0.944489	0.951485	0.946030	0.935423	0.933267	0.932656	0.938787	0.941648	0
4	0.561135	0.561135	0.561135	0.563294	0.565056	0.565056	0.566266	0.568564	0.570862	0
5	0.630525	0.632752	0.638669	0.646816	0.652649	0.655925	0.654333	0.650120	0.640460	0
6	0.225986	0.245332	0.306594	0.304438	0.287543	0.238678	0.209466	0.193071	0.185980	0
7	0.432715	0.443749	0.374935	0.339013	0.396952	0.445157	0.404925	0.364693	0.381725	0
8	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1
9	0.763658	0.763658	0.763658	0.763658	0.763658	0.763658	0.763658	0.763658	0.763658	0
0	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1
1	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1
2	0.911239	0.910965	0.908327	0.909763	0.913341	0.920210	0.928124	0.935630	0.943071	0
3	0.762583	0.762791	0.764226	0.764900	0.765274	0.767148	0.769255	0.770306	0.771415	0
4	0.339931	0.339931	0.339931	0.339931	0.339931	0.339931	0.339931	0.339931	0.339931	0
5	0.811747	0.811747	0.811747	0.811747	0.811747	0.811747	0.811747	0.811747	0.811747	0
6	0.968123	0.967495	0.968642	0.969790	0.970424	0.970817	0.971157	0.971157	0.971157	0
7	0.540873	0.540873	0.540873	0.540873	0.540873	0.540873	0.540873	0.540873	0.540873	0
8	0.996451	0.994424	0.985269	0.980963	0.982586	0.985624	0.984999	0.979978	0.981284	0
9	0.421518	0.423566	0.418915	0.398364	0.373263	0.360707	0.372740	0.397506	0.399863	0
0	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1
1	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1
2	0.659031	0.659031	0.659031	0.659170	0.660769	0.662368	0.662670	0.662670	0.663593	0
3	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1
4	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1
5	0.949020	0.949020	0.949020	0.949020	0.949020	0.949020	0.949020	0.949020	0.949020	0
6	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1
7	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1
8	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1
9	0.604165	0.603629	0.605322	0.606972	0.606972	0.606972	0.608171	0.609864	0.611556	0

30 rows × 262145 columns

In [64]:

```
from sklearn.utils import shuffle
celebrities_indexed=shuffle(actor_1).reset_index()
celebrities_indexed
```

Out[64]:

	index	0	1	2	3	4	5	6	7	
0	7	0.432715	0.443749	0.374935	0.339013	0.396952	0.445157	0.404925	0.364693	0.3
1	1	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.0
2	4	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.0
3	9	0.763658	0.763658	0.763658	0.763658	0.763658	0.763658	0.763658	0.763658	0.7
4	5	0.630525	0.632752	0.638669	0.646816	0.652649	0.655925	0.654333	0.650120	0.6
5	9	0.421518	0.423566	0.418915	0.398364	0.373263	0.360707	0.372740	0.397506	0.3
6	7	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.0
7	6	0.968123	0.967495	0.968642	0.969790	0.970424	0.970817	0.971157	0.971157	0.9
8	0	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.0
9	8	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.0
10	3	0.943523	0.944489	0.951485	0.946030	0.935423	0.933267	0.932656	0.938787	0.9
11	9	0.604165	0.603629	0.605322	0.606972	0.606972	0.606972	0.608171	0.609864	0.6
12	2	0.659031	0.659031	0.659031	0.659170	0.660769	0.662368	0.662670	0.662670	0.6
13	4	0.339931	0.339931	0.339931	0.339931	0.339931	0.339931	0.339931	0.339931	0.3
14	1	0.988550	0.988550	0.988550	0.988550	0.988550	0.988550	0.988550	0.988550	0.9
15	8	0.996451	0.994424	0.985269	0.980963	0.982586	0.985624	0.984999	0.979978	0.9
16	8	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.0
17	2	0.111776	0.111776	0.111776	0.111776	0.111776	0.111776	0.111776	0.111776	0.1
18	0	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.0
19	5	0.949020	0.949020	0.949020	0.949020	0.949020	0.949020	0.949020	0.949020	0.9
20	2	0.911239	0.910965	0.908327	0.909763	0.913341	0.920210	0.928124	0.935630	0.9
21	6	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.0
22	0	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.0
23	5	0.811747	0.811747	0.811747	0.811747	0.811747	0.811747	0.811747	0.811747	0.8
24	7	0.540873	0.540873	0.540873	0.540873	0.540873	0.540873	0.540873	0.540873	0.5
25	6	0.225986	0.245332	0.306594	0.304438	0.287543	0.238678	0.209466	0.193071	0.1
26	3	0.762583	0.762791	0.764226	0.764900	0.765274	0.767148	0.769255	0.770306	0.7
27	1	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.0
28	4	0.561135	0.561135	0.561135	0.563294	0.565056	0.565056	0.566266	0.568564	0.5
29	3	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.0

30 rows × 262146 columns

In [65]:

```
celebrities_actor_1=celebrities_indexed.drop(['index'],axis=1)
celebrities_actor_1
```

Out[65]:

	0	1	2	3	4	5	6	7	8
0	0.432715	0.443749	0.374935	0.339013	0.396952	0.445157	0.404925	0.364693	0.381725
1	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
2	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
3	0.763658	0.763658	0.763658	0.763658	0.763658	0.763658	0.763658	0.763658	0.763658
4	0.630525	0.632752	0.638669	0.646816	0.652649	0.655925	0.654333	0.650120	0.640460
5	0.421518	0.423566	0.418915	0.398364	0.373263	0.360707	0.372740	0.397506	0.399863
6	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
7	0.968123	0.967495	0.968642	0.969790	0.970424	0.970817	0.971157	0.971157	0.971157
8	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
9	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
10	0.943523	0.944489	0.951485	0.946030	0.935423	0.933267	0.932656	0.938787	0.941648
11	0.604165	0.603629	0.605322	0.606972	0.606972	0.606972	0.608171	0.609864	0.611556
12	0.659031	0.659031	0.659031	0.659170	0.660769	0.662368	0.662670	0.662670	0.663593
13	0.339931	0.339931	0.339931	0.339931	0.339931	0.339931	0.339931	0.339931	0.339931
14	0.988550	0.988550	0.988550	0.988550	0.988550	0.988550	0.988550	0.988550	0.988550
15	0.996451	0.994424	0.985269	0.980963	0.982586	0.985624	0.984999	0.979978	0.981284
16	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
17	0.111776	0.111776	0.111776	0.111776	0.111776	0.111776	0.111776	0.111776	0.111776
18	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
19	0.949020	0.949020	0.949020	0.949020	0.949020	0.949020	0.949020	0.949020	0.949020
20	0.911239	0.910965	0.908327	0.909763	0.913341	0.920210	0.928124	0.935630	0.943071
21	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
22	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
23	0.811747	0.811747	0.811747	0.811747	0.811747	0.811747	0.811747	0.811747	0.811747
24	0.540873	0.540873	0.540873	0.540873	0.540873	0.540873	0.540873	0.540873	0.540873
25	0.225986	0.245332	0.306594	0.304438	0.287543	0.238678	0.209466	0.193071	0.185980
26	0.762583	0.762791	0.764226	0.764900	0.765274	0.767148	0.769255	0.770306	0.771415
27	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
28	0.561135	0.561135	0.561135	0.563294	0.565056	0.565056	0.566266	0.568564	0.570862
29	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000

30 rows × 262145 columns

In [66]:

```
celebrities_actor_1.to_csv("C:/Users/SRIVENKATESH/Downloads/CEC.csv")
```

In [67]:

```
x=celebrities_actor_1.values[:,:-1]
y=celebrities_actor_1.values[:,-1]
```

In [68]:

```
x
```

Out[68]:

```
array([[0.4327151439898622, 0.4437485868723252, 0.3749347457369636, ...,
        0.5621746139705883, 0.5639898713235294, 0.5636988174019608],
       [1.0, 1.0, 1.0, ..., 1.0, 1.0, 1.0],
       [1.0, 1.0, 1.0, ..., 1.0, 1.0, 1.0],
       ...,
       [1.0, 1.0, 1.0, ..., 1.0, 1.0, 1.0],
       [0.5611345098039217, 0.5611345098039217, 0.5611345098039217, ...,
        0.46630876531862747, 0.46630876531862747, 0.4663087653186274],
       [1.0, 1.0, 1.0, ..., 1.0, 1.0, 1.0]], dtype=object)
```

In [69]:

```
y
```

Out[69]:

```
array(['ca', 'ch', 'ch', 'ca', 'ca', 'el', 'ch', 'el', 'el', 'ch', 'ca',
       'ch', 'ch', 'el', 'ca', 'el', 'ca', 'ca', 'ca', 'ch', 'el', 'ch',
       'ch', 'el', 'el', 'ca', 'el', 'el', 'ca', 'ch'], dtype=object)
```

In [75]:

```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=0)
```

In [76]:

```
from sklearn import svm
```

In [77]:

```
clf=svm.SVC()
clf.fit(x_train, y_train)
```

Out[77]:

```
SVC()
```

In [78]:

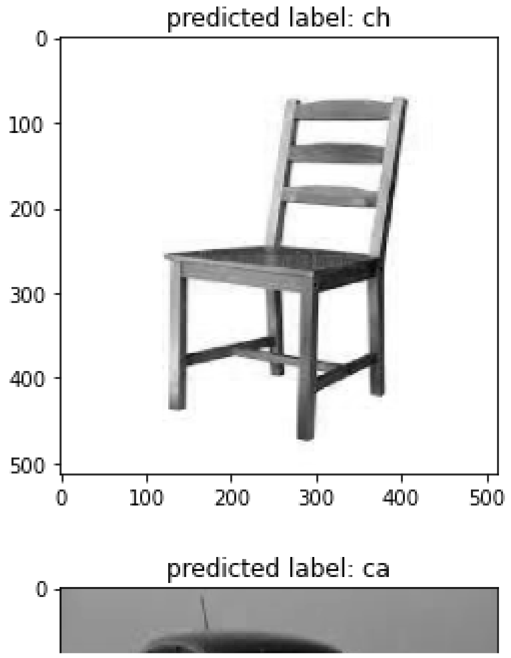
```
y_pred=clf.predict(x_test)
y_pred
```

Out[78]:

```
array(['ch', 'ca', 'ca', 'ch', 'ca', 'ca', 'ch', 'ch', 'ca', 'ch', 'ca',
       'ca', 'ca', 'ca', 'ch'], dtype=object)
```

In [80]:

```
for i in range(10):
  predicted_images=(np.reshape(x_test[i],(512,512)).astype(np.float64))
  plt.title('predicted label: {0}'. format(y_pred[i]))
  plt.imshow(predicted_images, interpolation='nearest', cmap='gray')
  plt.show()
```



In [81]:

```
from sklearn import metrics
accuracy=metrics.accuracy_score(y_test,y_pred)
accuracy
```

Out[81]:

```
0.4666666666666667
```

In [82]:

```
from sklearn.metrics import confusion_matrix
confusion_matrix(y_test,y_pred)
```

Out[82]:

```
array([[4, 1, 0],
       [0, 3, 0],
       [5, 2, 0]], dtype=int64)
```

In [ ]: