In [1]:

```
import os
```

In [2]:

```
import warnings
```

In [3]:

```
warnings.simplefilter('ignore')
```

In [4]:

```
import numpy as np
```

In [5]:

```
import pandas as pd
```

In [6]:

```
import matplotlib.pyplot as plt
```

In [7]:

```
%matplotlib inline
```

In [8]:

```
from skimage.io import imread, imshow
from skimage.transform import resize
from skimage.color import rgb2gray
```

In [15]:

```
vj=os.listdir("C:/Users/SRIVENKATESH/Downloads/ML NEW/VIJAY")
```

In [16]:

```
tz=os.listdir("C:/Users/SRIVENKATESH/Downloads/ML NEW/TZUYU")
```

In [17]:

```
ch=os.listdir("C:/Users/SRIVENKATESH/Downloads/ML NEW/CHRIS EVANS")
```

In [20]:

```
limit=30
vj_image=[None]*limit
j=0
for i in vj:
    if(j<limit):
        vj_image[j]=imread("C:/Users/SRIVENKATESH/Downloads/ML NEW/VIJAY/"+i)
        j+=1
    else:
        break
```

In [21]:

```
limit=30
tz_image=[None]*limit
j=0
for i in tz:
    if(j<limit):
        tz_image[j]=imread("C:/Users/SRIVENKATESH/Downloads/ML NEW/TZUYU/"+i)
        j+=1
    else:
        break
```

In [22]:

```
limit=30
ch_image=[None]*limit
j=0
for i in ch:
    if(j<limit):
        ch_image[j]=imread("C:/Users/SRIVENKATESH/Downloads/ML NEW/CHRIS EVANS/"+i)
        j+=1
    else:
        break
```

In [64]:

```
imshow(vj_image[24])
```

Out[64]:

```
<matplotlib.image.AxesImage at 0x2b6c76933a0>
```

In [63]:

```
imshow(tz_image[24])
```

Out[63]:

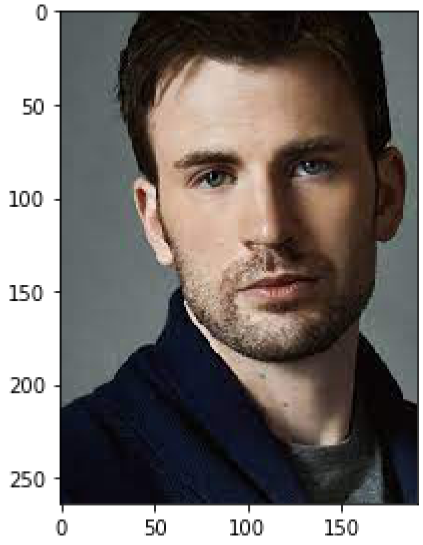
```
<matplotlib.image.AxesImage at 0x2b6c7626cd0>
```

In [65]:

```
imshow(ch_image[24])
```

Out[65]:

```
<matplotlib.image.AxesImage at 0x2b6c76f4fa0>
```



In [66]:

```
vj_image[24].shape
```

Out[66]:

```
(126, 126, 3)
```

In [67]:

```
tz_image[24].shape
```

Out[67]:

```
(275, 183, 3)
```

In [68]:

```
ch_image[24].shape
```

Out[68]:

```
(264, 191, 3)
```

In [69]:

```
vj_gray=[None]*limit
j=0
for i in vj:
    if(j<limit):
        vj_gray[j]=rgb2gray(vj_image[j])
        j+=1
    else:
        break
```

In [70]:

```
tz_gray=[None]*limit
j=0
for i in tz:
    if(j<limit):
        tz_gray[j]=rgb2gray(tz_image[j])
        j+=1
    else:
        break
```

In [71]:

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

In [72]:

```
imshow(vj_gray[24])
```

Out[72]:

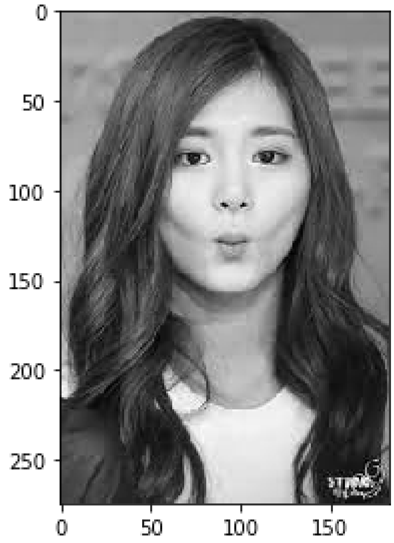
```
<matplotlib.image.AxesImage at 0x2b6c7760730>
```

In [73]:

```
imshow(tz_gray[24])
```

Out[73]:

```
<matplotlib.image.AxesImage at 0x2b6c77d1250>
```



In [74]:

```
imshow(ch_gray[24])
```

Out[74]:

```
<matplotlib.image.AxesImage at 0x2b6c7832820>
```

In [75]:

```
vj_gray[24].shape
```

Out[75]:

```
(126, 126)
```

In [76]:

```
tz_gray[24].shape
```

Out[76]:

```
(275, 183)
```

In [77]:

```
ch_gray[24].shape
```

Out[77]:

```
(264, 191)
```

In [78]:

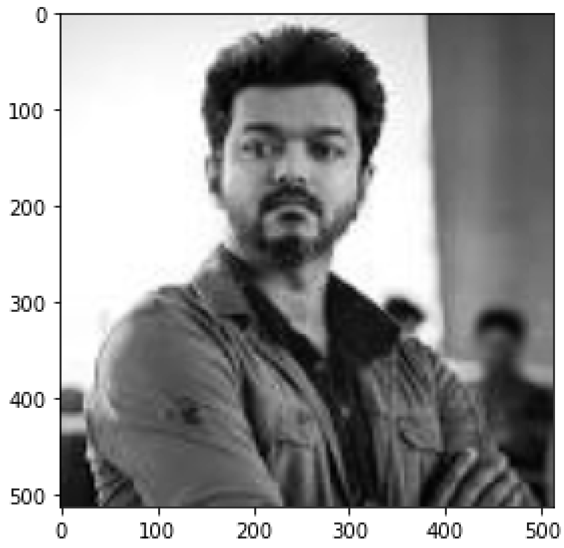
```
for j in range(30):
    vj=vj_gray[j]
    vj_gray[j]=resize(vj,(512,512))
for j in range(30):
    tz=tz_gray[j]
    tz_gray[j]=resize(tz,(512,512))
for j in range(30):
    ch=ch_gray[j]
    ch_gray[j]=resize(ch,(512,512))
```

In [79]:

```
imshow(vj_gray[24])
```

Out[79]:

```
<matplotlib.image.AxesImage at 0x2b6c8913f10>
```

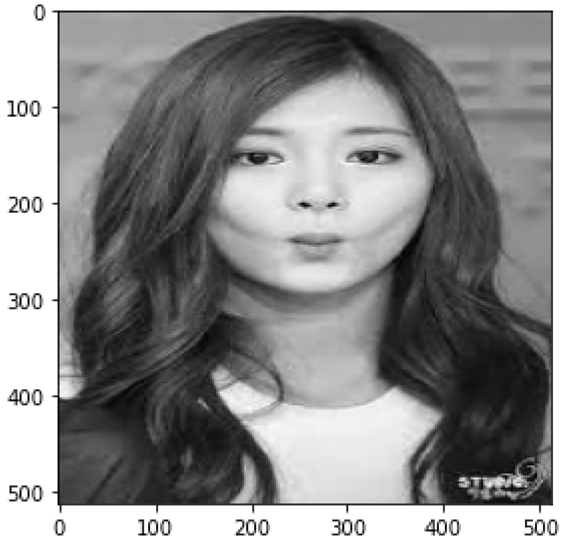


In [80]:

```
imshow(tz_gray[24])
```

Out[80]:

```
<matplotlib.image.AxesImage at 0x2b6c8977e80>
```

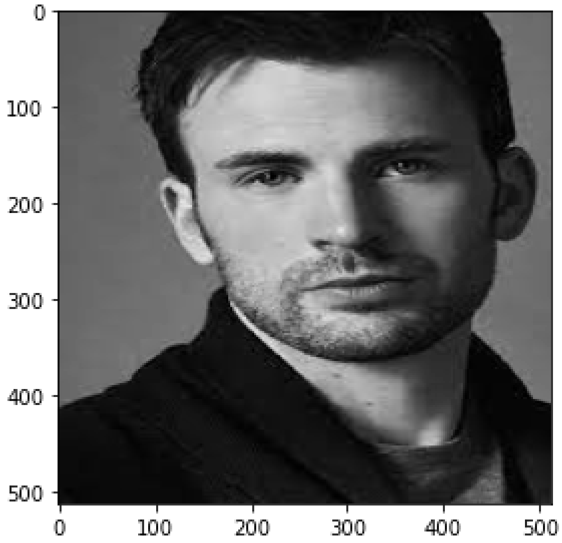


In [81]:

```
imshow(ch_gray[24])
```

Out[81]:

```
<matplotlib.image.AxesImage at 0x2b6c89dbe20>
```



In [82]:

```
len_of_image_vj=len(vj_gray)
len_of_image_vj
len_of_image_tz=len(tz_gray)
len_of_image_tz
len_of_image_ch=len(ch_gray)
len_of_image_ch
```

Out[82]:

```
30
```

In [83]:

```
image_size_vj=vj_gray[24].shape
image_size_vj
image_size_tz=tz_gray[24].shape
image_size_tz
image_size_ch=ch_gray[24].shape
image_size_ch
```

Out[83]:

```
(512, 512)
```

In [136]:

```
flatten_size_vj=image_size_vj[0]*image_size_vj[1]
flatten_size_vj
flatten_size_tz=image_size_tz[0]*image_size_tz[1]
flatten_size_tz
flatten_size_ch=image_size_ch[0]*image_size_ch[1]
flatten_size_ch
```

Out[136]:

```
262144
```

In [137]:

```
for i in range(len_of_image_vj):
    vj_gray[i]=np.ndarray.flatten(vj_gray[i]).reshape(flatten_size_vj,1)
vj_gray[24].shape
for i in range(len_of_image_tz):
    tz_gray[i]=np.ndarray.flatten(tz_gray[i]).reshape(flatten_size_tz,1)
tz_gray[24].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[24].shape
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Input In [137], in <cell line: 1>()
      1 for i in range(len_of_image_vj):
----> 2     vj_gray[i]=np.ndarray.flatten(vj_gray[i]).reshape(flatten_size_v
j,1)
      3 vj_gray[24].shape
      4 for i in range(len_of_image_tz):

ValueError: could not broadcast input array from shape (262144,1) into shape
(262144,)
```

In [138]:

```
vj_gray=np.dstack(vj_gray)
tz_gray=np.dstack(tz_gray)
ch_gray=np.dstack(ch_gray)
```

In [139]:

```
vj_gray=np.rollaxis(vj_gray,axis=2,start=0)
vj_gray.shape
tz_gray=np.rollaxis(tz_gray,axis=2,start=0)
tz_gray.shape
ch_gray=np.rollaxis(ch_gray,axis=2,start=0)
ch_gray.shape
```

Out[139]:

```
(30, 1, 262144)
```

In [140]:

```
vj_gray=vj_gray.reshape(len_of_image_vj,flatten_size_vj)
vj_gray.shape
tz_gray=tz_gray.reshape(len_of_image_tz,flatten_size_tz)
tz_gray.shape
ch_gray=ch_gray.reshape(len_of_image_ch,flatten_size_ch)
ch_gray.shape
```

Out[140]:

```
(30, 262144)
```

In [141]:

```
vj_data=pd.DataFrame(vj_gray)
vj_data
```

Out[141]:

	0	1	2	3	4	5	6	7	8
0	0.250464	0.250736	0.251008	0.251281	0.251549	0.251276	0.251004	0.250731	0.250459
1	0.970121	0.966849	0.963577	0.964999	0.968271	0.971543	0.974815	0.978087	0.980416
2	0.048640	0.047872	0.048978	0.050083	0.050843	0.051192	0.051542	0.053579	0.056140
3	0.787977	0.787977	0.787977	0.787764	0.787199	0.786410	0.785091	0.784056	0.784056
4	0.206417	0.254817	0.643440	0.818705	0.896679	0.903521	0.893824	0.909426	0.925271
5	0.087398	0.087398	0.087398	0.087398	0.087398	0.087398	0.087398	0.087398	0.087398
6	0.127296	0.127424	0.132987	0.136680	0.138748	0.142295	0.147014	0.153429	0.161066
7	0.668988	0.669705	0.668300	0.666895	0.665300	0.663638	0.661946	0.660151	0.658357
8	0.854494	0.854494	0.854494	0.854494	0.854494	0.854494	0.854494	0.854494	0.855628
9	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
10	0.412686	0.422629	0.419712	0.409092	0.410120	0.411956	0.411593	0.410233	0.408040
11	0.691842	0.690954	0.690065	0.690285	0.691173	0.692061	0.692950	0.693838	0.694726
12	0.328936	0.329275	0.330431	0.332660	0.334422	0.334422	0.334596	0.334927	0.336871
13	0.031588	0.031588	0.031588	0.031588	0.031588	0.031588	0.031588	0.031588	0.031588
14	0.413773	0.413231	0.414404	0.415577	0.415803	0.415803	0.415803	0.415803	0.415803
15	0.207322	0.202297	0.207628	0.212959	0.218085	0.221039	0.223994	0.225799	0.220757
16	0.341779	0.343250	0.340563	0.337876	0.338036	0.339540	0.342116	0.351745	0.361375
17	0.778521	0.828342	0.660008	0.506354	0.599929	0.693503	0.783519	0.872412	0.926989
18	0.396617	0.395888	0.394296	0.392867	0.390482	0.387950	0.391231	0.393471	0.394742
19	0.506437	0.506685	0.506738	0.504397	0.504449	0.505854	0.505854	0.505292	0.504207
20	0.454902	0.454902	0.454902	0.454902	0.455404	0.457694	0.458824	0.458824	0.458824
21	0.128174	0.129457	0.133451	0.137892	0.143234	0.150624	0.156937	0.162602	0.170158
22	0.992157	0.992157	0.992157	0.992157	0.992157	0.992157	0.992157	0.992157	0.992157
23	0.894704	0.894704	0.894704	0.894704	0.894704	0.894704	0.894704	0.894704	0.894704
24	0.880404	0.880404	0.880404	0.880404	0.880404	0.880404	0.880551	0.880915	0.881278
25	0.056610	0.058020	0.059430	0.059365	0.057955	0.056545	0.055135	0.053725	0.052315
26	0.269302	0.268651	0.268000	0.267349	0.267991	0.268642	0.269293	0.269944	0.270595
27	0.870732	0.872754	0.874776	0.876798	0.876308	0.874286	0.872264	0.870242	0.868220
28	0.460929	0.459504	0.458080	0.458784	0.460209	0.461634	0.463058	0.464483	0.466080
29	0.027417	0.028917	0.030416	0.031016	0.029516	0.028017	0.026518	0.025018	0.023519

30 rows × 262144 columns

In [142]:

```
tz_data=pd.DataFrame(tz_gray)
tz_data
```

Out[142]:

	0	1	2	3	4	5	6	7	8
0	0.955769	0.955769	0.955769	0.955769	0.955769	0.955769	0.955769	0.955769	0.955769
1	0.524310	0.525709	0.520815	0.516018	0.512489	0.508961	0.503202	0.496835	0.493670
2	0.620060	0.620060	0.620060	0.620060	0.620060	0.620060	0.620060	0.620060	0.620060
3	0.227086	0.227086	0.227086	0.227086	0.227086	0.227086	0.227086	0.227086	0.227086
4	0.997249	0.996771	0.993311	0.995224	0.999361	1.000000	1.000000	0.984831	0.966487
5	0.196665	0.192893	0.206581	0.217639	0.200990	0.184342	0.151924	0.115835	0.097395
6	0.745270	0.748442	0.742646	0.736850	0.733186	0.730530	0.728741	0.732665	0.736589
7	0.484325	0.484325	0.484325	0.484325	0.484325	0.484594	0.484867	0.485140	0.486147
8	0.738427	0.742383	0.737425	0.732467	0.728286	0.725888	0.723491	0.726814	0.735735
9	0.791977	0.791256	0.792573	0.793891	0.793535	0.792388	0.792001	0.796619	0.801238
10	0.953437	0.951670	0.950538	0.950201	0.947153	0.944964	0.939569	0.939569	0.939569
11	0.525220	0.530952	0.521984	0.513017	0.508572	0.507960	0.507348	0.513690	0.520787
12	0.762247	0.762149	0.762323	0.762497	0.763283	0.764385	0.765629	0.768152	0.770675
13	0.555201	0.555244	0.555181	0.555117	0.559514	0.568762	0.578009	0.593334	0.610194
14	0.049306	0.049306	0.049306	0.049306	0.049306	0.049306	0.049306	0.049306	0.049306
15	0.905377	0.905377	0.905377	0.905377	0.905377	0.905377	0.905377	0.905377	0.905377
16	0.672212	0.671996	0.672390	0.672784	0.673956	0.675496	0.677035	0.678575	0.680114
17	0.402956	0.402956	0.402956	0.401213	0.399307	0.397234	0.395042	0.392744	0.390308
18	0.983730	0.984557	0.983010	0.981463	0.977755	0.973114	0.968293	0.962554	0.956814
19	0.679618	0.679438	0.679697	0.679955	0.678908	0.676315	0.673722	0.672460	0.671611
20	0.755816	0.755189	0.756336	0.757483	0.757851	0.757851	0.757817	0.757555	0.757293
21	0.344516	0.344516	0.344516	0.344516	0.344516	0.344516	0.344516	0.344516	0.344516
22	0.506798	0.506798	0.506798	0.506798	0.506798	0.506798	0.506798	0.506798	0.506798
23	0.875588	0.875588	0.875588	0.875588	0.875588	0.875588	0.875588	0.875588	0.875588
24	0.629770	0.629770	0.629770	0.629770	0.629770	0.629770	0.629770	0.630479	0.631880
25	0.878431	0.878431	0.878431	0.878431	0.878431	0.878431	0.878431	0.878431	0.878431
26	0.477538	0.476900	0.477700	0.478501	0.479059	0.479059	0.479059	0.479008	0.478906
27	0.924598	0.946623	0.933702	0.911676	0.889650	0.871784	0.860704	0.849623	0.838543
28	0.538641	0.538395	0.538742	0.539088	0.539446	0.539820	0.540193	0.541348	0.542795
29	0.331125	0.331125	0.331125	0.331125	0.331125	0.331125	0.331125	0.331125	0.331125

30 rows × 262144 columns

In [143]:

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

Out[143]:

	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	0.528654	0.528654	0.528654	0.528654	0.528654	0.528654	0.528252	0.526681	0.525111
2	0.044960	0.043939	0.044911	0.045931	0.046952	0.046972	0.046916	0.046859	0.046896
3	0.672140	0.672140	0.672140	0.672140	0.672140	0.672140	0.672140	0.670915	0.669444
4	0.905840	0.905840	0.905840	0.905840	0.905840	0.905840	0.905840	0.905840	0.905840
5	0.028277	0.028277	0.028277	0.028277	0.028277	0.028277	0.028978	0.030594	0.032199
6	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
7	0.847432	0.847432	0.847432	0.847432	0.847432	0.847432	0.847432	0.847432	0.847432
8	0.717853	0.717853	0.717853	0.717853	0.717853	0.717853	0.717853	0.718319	0.719031
9	0.369490	0.368908	0.364695	0.357981	0.350233	0.340214	0.329780	0.318047	0.306535
10	0.708208	0.706492	0.709628	0.712763	0.716906	0.721524	0.725940	0.729019	0.732098
11	0.795773	0.795773	0.795773	0.795773	0.795505	0.795112	0.794619	0.793473	0.792327
12	0.151855	0.151855	0.151855	0.153103	0.155141	0.155776	0.155776	0.156267	0.156910
13	0.910190	0.910481	0.912588	0.913204	0.913204	0.908886	0.903778	0.897479	0.890684
14	0.566525	0.567976	0.562711	0.558009	0.559241	0.560473	0.566996	0.574753	0.577007
15	0.187938	0.187938	0.187938	0.187938	0.187938	0.187938	0.187938	0.187938	0.187938
16	0.506877	0.506801	0.506953	0.507106	0.507451	0.507858	0.508422	0.509438	0.510454
17	0.864023	0.864023	0.864023	0.864023	0.864023	0.864023	0.864023	0.864023	0.864023
18	0.299314	0.301478	0.297865	0.294253	0.290570	0.286839	0.283113	0.280678	0.278244
19	0.666711	0.666711	0.666711	0.666711	0.666711	0.666711	0.666711	0.666711	0.666711
20	0.790552	0.790293	0.790617	0.790942	0.791494	0.792572	0.793649	0.794890	0.796292
21	0.758112	0.758112	0.758112	0.758112	0.758112	0.758112	0.758112	0.758112	0.758112
22	0.003922	0.003922	0.003922	0.003922	0.003922	0.003922	0.003922	0.003922	0.003922
23	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
24	0.346840	0.346840	0.346840	0.346840	0.347541	0.349004	0.350467	0.350762	0.350762
25	0.910994	0.916903	0.909496	0.902089	0.885572	0.848124	0.810675	0.780546	0.757577
26	0.964706	0.964706	0.964706	0.964706	0.964706	0.964706	0.964706	0.964706	0.964706
27	0.285585	0.285585	0.285585	0.285436	0.283713	0.281989	0.280266	0.278543	0.276819
28	0.092168	0.092168	0.092168	0.092168	0.092168	0.092168	0.092168	0.092168	0.092168
29	0.564222	0.547465	0.490340	0.395612	0.313339	0.279874	0.346190	0.502182	0.586132

30 rows × 262144 columns

In [144]:

```
vj_data["label"]="vj"
vj_data
```

Out[144]:

	0	1	2	3	4	5	6	7	8
0	0.250464	0.250736	0.251008	0.251281	0.251549	0.251276	0.251004	0.250731	0.250459
1	0.970121	0.966849	0.963577	0.964999	0.968271	0.971543	0.974815	0.978087	0.980416
2	0.048640	0.047872	0.048978	0.050083	0.050843	0.051192	0.051542	0.053579	0.056140
3	0.787977	0.787977	0.787977	0.787764	0.787199	0.786410	0.785091	0.784056	0.784056
4	0.206417	0.254817	0.643440	0.818705	0.896679	0.903521	0.893824	0.909426	0.925271
5	0.087398	0.087398	0.087398	0.087398	0.087398	0.087398	0.087398	0.087398	0.087398
6	0.127296	0.127424	0.132987	0.136680	0.138748	0.142295	0.147014	0.153429	0.161066
7	0.668988	0.669705	0.668300	0.666895	0.665300	0.663638	0.661946	0.660151	0.658357
8	0.854494	0.854494	0.854494	0.854494	0.854494	0.854494	0.854494	0.854494	0.855628
9	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
10	0.412686	0.422629	0.419712	0.409092	0.410120	0.411956	0.411593	0.410233	0.408040
11	0.691842	0.690954	0.690065	0.690285	0.691173	0.692061	0.692950	0.693838	0.694726
12	0.328936	0.329275	0.330431	0.332660	0.334422	0.334422	0.334596	0.334927	0.336871
13	0.031588	0.031588	0.031588	0.031588	0.031588	0.031588	0.031588	0.031588	0.031588
14	0.413773	0.413231	0.414404	0.415577	0.415803	0.415803	0.415803	0.415803	0.415803
15	0.207322	0.202297	0.207628	0.212959	0.218085	0.221039	0.223994	0.225799	0.220757
16	0.341779	0.343250	0.340563	0.337876	0.338036	0.339540	0.342116	0.351745	0.361375
17	0.778521	0.828342	0.660008	0.506354	0.599929	0.693503	0.783519	0.872412	0.926989
18	0.396617	0.395888	0.394296	0.392867	0.390482	0.387950	0.391231	0.393471	0.394742
19	0.506437	0.506685	0.506738	0.504397	0.504449	0.505854	0.505854	0.505292	0.504207
20	0.454902	0.454902	0.454902	0.454902	0.455404	0.457694	0.458824	0.458824	0.458824
21	0.128174	0.129457	0.133451	0.137892	0.143234	0.150624	0.156937	0.162602	0.170158
22	0.992157	0.992157	0.992157	0.992157	0.992157	0.992157	0.992157	0.992157	0.992157
23	0.894704	0.894704	0.894704	0.894704	0.894704	0.894704	0.894704	0.894704	0.894704
24	0.880404	0.880404	0.880404	0.880404	0.880404	0.880404	0.880551	0.880915	0.881278
25	0.056610	0.058020	0.059430	0.059365	0.057955	0.056545	0.055135	0.053725	0.052315
26	0.269302	0.268651	0.268000	0.267349	0.267991	0.268642	0.269293	0.269944	0.270595
27	0.870732	0.872754	0.874776	0.876798	0.876308	0.874286	0.872264	0.870242	0.868220
28	0.460929	0.459504	0.458080	0.458784	0.460209	0.461634	0.463058	0.464483	0.466080
29	0.027417	0.028917	0.030416	0.031016	0.029516	0.028017	0.026518	0.025018	0.023519

30 rows × 262145 columns

In [145]:

```
tz_data["label"]="tz"
tz_data
```

Out[145]:

	0	1	2	3	4	5	6	7	8
0	0.955769	0.955769	0.955769	0.955769	0.955769	0.955769	0.955769	0.955769	0.955769
1	0.524310	0.525709	0.520815	0.516018	0.512489	0.508961	0.503202	0.496835	0.493670
2	0.620060	0.620060	0.620060	0.620060	0.620060	0.620060	0.620060	0.620060	0.620060
3	0.227086	0.227086	0.227086	0.227086	0.227086	0.227086	0.227086	0.227086	0.227086
4	0.997249	0.996771	0.993311	0.995224	0.999361	1.000000	1.000000	0.984831	0.966487
5	0.196665	0.192893	0.206581	0.217639	0.200990	0.184342	0.151924	0.115835	0.097395
6	0.745270	0.748442	0.742646	0.736850	0.733186	0.730530	0.728741	0.732665	0.736589
7	0.484325	0.484325	0.484325	0.484325	0.484325	0.484594	0.484867	0.485140	0.486147
8	0.738427	0.742383	0.737425	0.732467	0.728286	0.725888	0.723491	0.726814	0.735735
9	0.791977	0.791256	0.792573	0.793891	0.793535	0.792388	0.792001	0.796619	0.801238
10	0.953437	0.951670	0.950538	0.950201	0.947153	0.944964	0.939569	0.939569	0.939569
11	0.525220	0.530952	0.521984	0.513017	0.508572	0.507960	0.507348	0.513690	0.520787
12	0.762247	0.762149	0.762323	0.762497	0.763283	0.764385	0.765629	0.768152	0.770675
13	0.555201	0.555244	0.555181	0.555117	0.559514	0.568762	0.578009	0.593334	0.610194
14	0.049306	0.049306	0.049306	0.049306	0.049306	0.049306	0.049306	0.049306	0.049306
15	0.905377	0.905377	0.905377	0.905377	0.905377	0.905377	0.905377	0.905377	0.905377
16	0.672212	0.671996	0.672390	0.672784	0.673956	0.675496	0.677035	0.678575	0.680114
17	0.402956	0.402956	0.402956	0.401213	0.399307	0.397234	0.395042	0.392744	0.390308
18	0.983730	0.984557	0.983010	0.981463	0.977755	0.973114	0.968293	0.962554	0.956814
19	0.679618	0.679438	0.679697	0.679955	0.678908	0.676315	0.673722	0.672460	0.671611
20	0.755816	0.755189	0.756336	0.757483	0.757851	0.757851	0.757817	0.757555	0.757293
21	0.344516	0.344516	0.344516	0.344516	0.344516	0.344516	0.344516	0.344516	0.344516
22	0.506798	0.506798	0.506798	0.506798	0.506798	0.506798	0.506798	0.506798	0.506798
23	0.875588	0.875588	0.875588	0.875588	0.875588	0.875588	0.875588	0.875588	0.875588
24	0.629770	0.629770	0.629770	0.629770	0.629770	0.629770	0.629770	0.630479	0.631880
25	0.878431	0.878431	0.878431	0.878431	0.878431	0.878431	0.878431	0.878431	0.878431
26	0.477538	0.476900	0.477700	0.478501	0.479059	0.479059	0.479059	0.479008	0.478906
27	0.924598	0.946623	0.933702	0.911676	0.889650	0.871784	0.860704	0.849623	0.838543
28	0.538641	0.538395	0.538742	0.539088	0.539446	0.539820	0.540193	0.541348	0.542795
29	0.331125	0.331125	0.331125	0.331125	0.331125	0.331125	0.331125	0.331125	0.331125

30 rows × 262145 columns

In [146]:

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

Out[146]:

	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	0.528654	0.528654	0.528654	0.528654	0.528654	0.528654	0.528252	0.526681	0.525111
2	0.044960	0.043939	0.044911	0.045931	0.046952	0.046972	0.046916	0.046859	0.046896
3	0.672140	0.672140	0.672140	0.672140	0.672140	0.672140	0.672140	0.670915	0.669444
4	0.905840	0.905840	0.905840	0.905840	0.905840	0.905840	0.905840	0.905840	0.905840
5	0.028277	0.028277	0.028277	0.028277	0.028277	0.028277	0.028978	0.030594	0.032199
6	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
7	0.847432	0.847432	0.847432	0.847432	0.847432	0.847432	0.847432	0.847432	0.847432
8	0.717853	0.717853	0.717853	0.717853	0.717853	0.717853	0.717853	0.718319	0.719031
9	0.369490	0.368908	0.364695	0.357981	0.350233	0.340214	0.329780	0.318047	0.306535
10	0.708208	0.706492	0.709628	0.712763	0.716906	0.721524	0.725940	0.729019	0.732098
11	0.795773	0.795773	0.795773	0.795773	0.795505	0.795112	0.794619	0.793473	0.792327
12	0.151855	0.151855	0.151855	0.153103	0.155141	0.155776	0.155776	0.156267	0.156910
13	0.910190	0.910481	0.912588	0.913204	0.913204	0.908886	0.903778	0.897479	0.890684
14	0.566525	0.567976	0.562711	0.558009	0.559241	0.560473	0.566996	0.574753	0.577007
15	0.187938	0.187938	0.187938	0.187938	0.187938	0.187938	0.187938	0.187938	0.187938
16	0.506877	0.506801	0.506953	0.507106	0.507451	0.507858	0.508422	0.509438	0.510454
17	0.864023	0.864023	0.864023	0.864023	0.864023	0.864023	0.864023	0.864023	0.864023
18	0.299314	0.301478	0.297865	0.294253	0.290570	0.286839	0.283113	0.280678	0.278244
19	0.666711	0.666711	0.666711	0.666711	0.666711	0.666711	0.666711	0.666711	0.666711
20	0.790552	0.790293	0.790617	0.790942	0.791494	0.792572	0.793649	0.794890	0.796292
21	0.758112	0.758112	0.758112	0.758112	0.758112	0.758112	0.758112	0.758112	0.758112
22	0.003922	0.003922	0.003922	0.003922	0.003922	0.003922	0.003922	0.003922	0.003922
23	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
24	0.346840	0.346840	0.346840	0.346840	0.347541	0.349004	0.350467	0.350762	0.350762
25	0.910994	0.916903	0.909496	0.902089	0.885572	0.848124	0.810675	0.780546	0.757577
26	0.964706	0.964706	0.964706	0.964706	0.964706	0.964706	0.964706	0.964706	0.964706
27	0.285585	0.285585	0.285585	0.285436	0.283713	0.281989	0.280266	0.278543	0.276819
28	0.092168	0.092168	0.092168	0.092168	0.092168	0.092168	0.092168	0.092168	0.092168
29	0.564222	0.547465	0.490340	0.395612	0.313339	0.279874	0.346190	0.502182	0.586132

30 rows × 262145 columns

In [147]:

```
actor_1=pd.concat([vj_data,tz_data,ch_data])
actor_1
```

Out[147]:

	0	1	2	3	4	5	6	7	8
0	0.250464	0.250736	0.251008	0.251281	0.251549	0.251276	0.251004	0.250731	0.250459
1	0.970121	0.966849	0.963577	0.964999	0.968271	0.971543	0.974815	0.978087	0.980416
2	0.048640	0.047872	0.048978	0.050083	0.050843	0.051192	0.051542	0.053579	0.056140
3	0.787977	0.787977	0.787977	0.787764	0.787199	0.786410	0.785091	0.784056	0.784056
4	0.206417	0.254817	0.643440	0.818705	0.896679	0.903521	0.893824	0.909426	0.925271
...	...	...	...	...	...	...	...	...	...
25	0.910994	0.916903	0.909496	0.902089	0.885572	0.848124	0.810675	0.780546	0.757577
26	0.964706	0.964706	0.964706	0.964706	0.964706	0.964706	0.964706	0.964706	0.964706
27	0.285585	0.285585	0.285585	0.285436	0.283713	0.281989	0.280266	0.278543	0.276819
28	0.092168	0.092168	0.092168	0.092168	0.092168	0.092168	0.092168	0.092168	0.092168
29	0.564222	0.547465	0.490340	0.395612	0.313339	0.279874	0.346190	0.502182	0.586132

90 rows × 262145 columns

In [148]:

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

Out[148]:

	index	0	1	2	3	4	5	6	7	
0	6	0.127296	0.127424	0.132987	0.136680	0.138748	0.142295	0.147014	0.153429	0.1
1	21	0.758112	0.758112	0.758112	0.758112	0.758112	0.758112	0.758112	0.758112	0.7
2	11	0.525220	0.530952	0.521984	0.513017	0.508572	0.507960	0.507348	0.513690	0.5
3	23	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.0
4	24	0.880404	0.880404	0.880404	0.880404	0.880404	0.880404	0.880551	0.880915	0.8
...	...	...	...	...	...	...	...	...	...	
85	14	0.413773	0.413231	0.414404	0.415577	0.415803	0.415803	0.415803	0.415803	0.4
86	19	0.666711	0.666711	0.666711	0.666711	0.666711	0.666711	0.666711	0.666711	0.6
87	8	0.738427	0.742383	0.737425	0.732467	0.728286	0.725888	0.723491	0.726814	0.7
88	7	0.668988	0.669705	0.668300	0.666895	0.665300	0.663638	0.661946	0.660151	0.6
89	12	0.151855	0.151855	0.151855	0.153103	0.155141	0.155776	0.155776	0.156267	0.1

90 rows × 262146 columns

In [149]:

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

Out[149]:

	0	1	2	3	4	5	6	7	8
0	0.127296	0.127424	0.132987	0.136680	0.138748	0.142295	0.147014	0.153429	0.161066
1	0.758112	0.758112	0.758112	0.758112	0.758112	0.758112	0.758112	0.758112	0.758112
2	0.525220	0.530952	0.521984	0.513017	0.508572	0.507960	0.507348	0.513690	0.520787
3	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
4	0.880404	0.880404	0.880404	0.880404	0.880404	0.880404	0.880551	0.880915	0.881278
...	...	...	...	...	...	...	...	...	...
85	0.413773	0.413231	0.414404	0.415577	0.415803	0.415803	0.415803	0.415803	0.415803
86	0.666711	0.666711	0.666711	0.666711	0.666711	0.666711	0.666711	0.666711	0.666711
87	0.738427	0.742383	0.737425	0.732467	0.728286	0.725888	0.723491	0.726814	0.735735
88	0.668988	0.669705	0.668300	0.666895	0.665300	0.663638	0.661946	0.660151	0.658357
89	0.151855	0.151855	0.151855	0.153103	0.155141	0.155776	0.155776	0.156267	0.156910

90 rows × 262145 columns

In [150]:

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

In [151]:

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

In [152]:

```
x
```

Out[152]:

```
array([[0.1272955372290518, 0.12742439321480545, 0.13298667659983915,
        ..., 0.18345118259803922, 0.18543494485294118,
        0.18548090073529408],
       [0.7581117647058824, 0.7581117647058823, 0.7581117647058823, ...,
        0.09019607843137256, 0.09019607843137256, 0.09019607843137256],
       [0.5252201513193168, 0.5309520705518536, 0.5219843904622395, ...,
        0.7673850850303501, 0.7685038684201709, 0.767788769758636],
       ...,
       [0.7384273583625344, 0.742382765598671, 0.737424960638308, ...,
        0.523443425245098, 0.5066235723039215, 0.5200426899509804],
       [0.6689883073754405, 0.6697045601040709, 0.6682996028286804, ...,
        0.7907214966299019, 0.7948191528799018, 0.7927301516544117],
       [0.15185490196078433, 0.15185490196078436, 0.15185490196078436,
        ..., 0.2426164705882353, 0.2426164705882353, 0.2426164705882353]],
      dtype=object)
```

In [153]:

```
y
```

Out[153]:

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

In [197]:

```
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 [198]:

```
from sklearn import svm
```

In [199]:

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

Out[199]:

```
SVC()
```

In [200]:

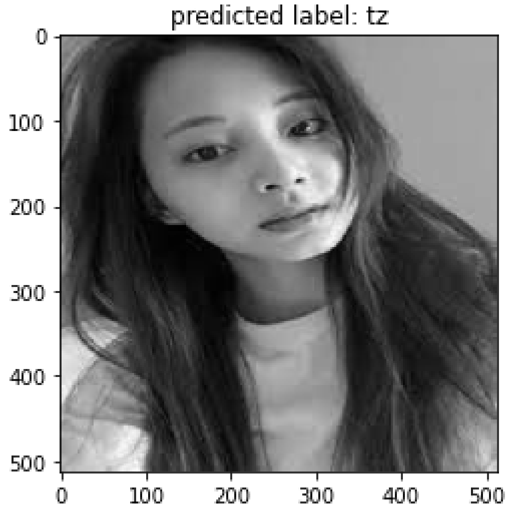
```
y_pred=clf.predict(x_test)
y_pred
```

Out[200]:

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

In [208]:

```
for i in range(15):
  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 [209]:

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

Out[209]:

```
0.6666666666666666
```

In [210]:

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

Out[210]:

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

In [ ]: