RECOGNITION IMAGE USING SVM

Image recognition

In [2]:

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

In [3]:

```
import numpy as np
import pandas as pd
```

In [4]:

```
import matplotlib.pyplot as plt
%matplotlib inline
```

In [5]:

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

In [6]:

```
ADHI=os.listdir("C:/Users/Magesh Kumar/Documents/IMAGE PROCESSING/ADHI")
```

In [7]:

```
VIJAY=os.listdir("C:/Users/Magesh Kumar/Documents/IMAGE PROCESSING/VIJAY SETHUPATHI")
```

In [8]:

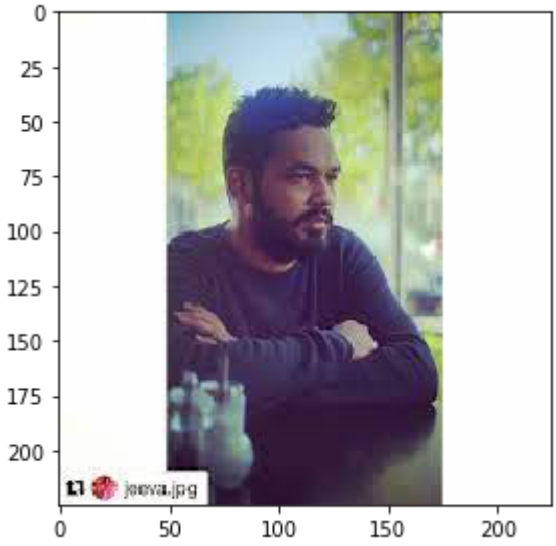
```
VISHAL=os.listdir("C:/Users/Magesh Kumar/Documents/IMAGE PROCESSING/VISHAL")
```

In [9]:

```
limit=10
ADHI_images=[None]*limit
j=0
for i in ADHI:
    if(j<limit):
        ADHI_images[j]=imread("C:/Users/Magesh Kumar/Documents/IMAGE PROCESSING/ADHI/"+i)
        j+=1
    else:
        break
imshow(ADHI_images[1])
```

Out[9]:

```
<matplotlib.image.AxesImage at 0x26fbad79070>
```



In [10]:

```
limit=10
VIJAY_images=[None]*limit
j=0
for i in VIJAY:
    if(j<limit):
        VIJAY_images[j]=imread("C:/Users/Magesh Kumar/Documents/IMAGE PROCESSING/VIJAY SETH
        j+=1
    else:
        break
imshow(VIJAY_images[1])
```

Out[10]:

```
<matplotlib.image.AxesImage at 0x26fbae39730>
```

In [11]:

```
limit=10
VISHAL_images=[None]*limit
j=0
for i in VISHAL:
    if(j<limit):
        VISHAL_images[j]=imread("C:/Users/Magesh Kumar/Documents/IMAGE PROCESSING/VISHAL/"+
        j+=1
    else:
        break
imshow(VISHAL_images[1])
```

Out[11]:

```
<matplotlib.image.AxesImage at 0x26fbaeaa850>
```

In [12]:

```
ADHI_images[1].shape
```

Out[12]:

```
(225, 225, 3)
```

In [13]:

```
VIJAY_images[1].shape
```

Out[13]:

```
(259, 194, 3)
```

In [14]:

```
VISHAL_images[1].shape
```

Out[14]:

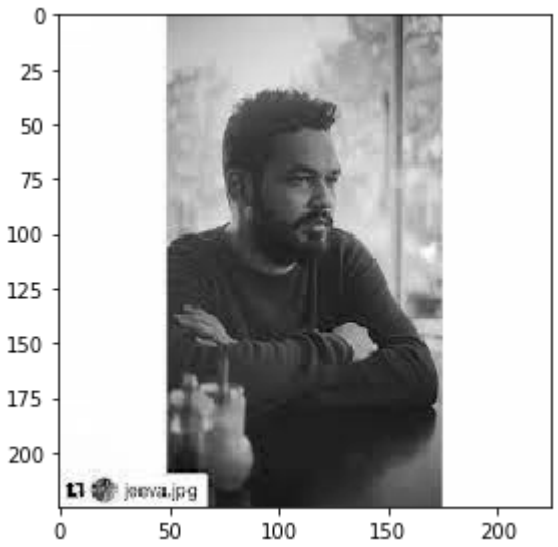
```
(275, 183, 3)
```

In [15]:

```
ADHI_gray=[None]*limit
j=0
for i in ADHI:
    if(j<limit):
        ADHI_gray[j]=rgb2gray(ADHI_images[j])
        j+=1
    else:
        break
imshow(ADHI_gray[1])
```

Out[15]:

```
<matplotlib.image.AxesImage at 0x26fbaf0fdc0>
```

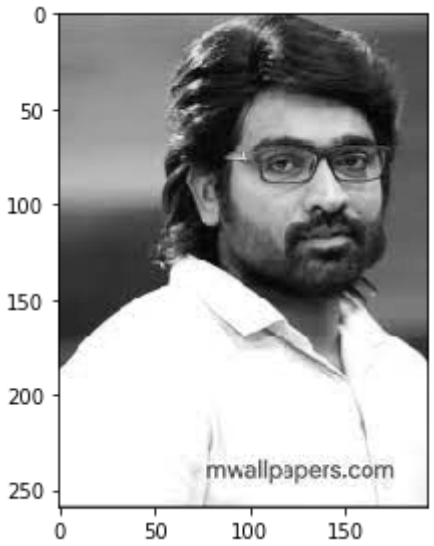


In [16]:

```
VIJAY_gray=[None]*limit
j=0
for i in VIJAY:
    if(j<limit):
        VIJAY_gray[j]=rgb2gray(VIJAY_images[j])
        j+=1
    else:
        break
imshow(VIJAY_gray[1])
```

Out[16]:

```
<matplotlib.image.AxesImage at 0x26fbbf73cd0>
```



In [17]:

```
VISHAL_gray=[None]*limit
j=0
for i in VISHAL:
    if(j<limit):
        VISHAL_gray[j]=rgb2gray(VISHAL_images[j])
        j+=1
    else:
        break
imshow(VISHAL_gray[1])
```

Out[17]:

```
<matplotlib.image.AxesImage at 0x26fbbfe32e0>
```

In [18]:

```
ADHI_gray[1].shape
```

Out[18]:

```
(225, 225)
```

In [19]:

```
VIJAY_gray[1].shape
```

Out[19]:

```
(259, 194)
```

In [20]:

```
VISHAL_gray[1].shape
```

Out[20]:

```
(275, 183)
```

In [21]:

```
for j in range(10):
    AD=ADHI_gray[j]
    ADHI_gray[j]=resize(AD,(512,512))
```

In [22]:

```
for j in range(10):
    VS=VIJAY_gray[j]
    VIJAY_gray[j]=resize(VS,(512,512))
```

In [23]:

```
for j in range(10):
    VK=VISHAL_gray[j]
    VISHAL_gray[j]=resize(VK,(512,512))
```

In [24]:

```
len_of_images_ADHI=len(ADHI_gray)
len_of_images_VIJAY=len(VIJAY_gray)
len_of_images_VISHAL=len(VISHAL_gray)
```

In [25]:

```
image_size_ADHI=ADHI_gray[1].shape
image_size_VIJAY=VIJAY_gray[1].shape
image_size_VISHAL=VISHAL_gray[1].shape
```

In [26]:

```
image_size_ADHI
image_size_VIJAY
image_size_VISHAL
```

Out[26]:

```
(512, 512)
```

In [27]:

```
flatten_size_ADHI=image_size_ADHI[0]*image_size_ADHI[1]
flatten_size_VIJAY=image_size_VIJAY[0]*image_size_VIJAY[1]
flatten_size_VISHAL=image_size_VISHAL[0]*image_size_VISHAL[1]
```

In [28]:

```
flatten_size_ADHI
```

Out[28]:

```
262144
```

In [29]:

```
flatten_size_VIJAY
```

Out[29]:

```
262144
```

In [30]:

```
flatten_size_VISHAL
```

Out[30]:

```
262144
```

In [31]:

```
for i in range(len_of_images_ADHI):
    ADHI_gray[i]=np.ndarray.flatten(ADHI_gray[i]).reshape(flatten_size_ADHI,1)
```

In [32]:

```
for i in range(len_of_images_VIJAY):
    VIJAY_gray[i]=np.ndarray.flatten(VIJAY_gray[i]).reshape(flatten_size_VIJAY,1)
```

In [33]:

```
for i in range(len_of_images_VISHAL):
    VISHAL_gray[i]=np.ndarray.flatten(VISHAL_gray[i]).reshape(flatten_size_VISHAL,1)
```

In [34]:

```
ADHI_gray=np.dstack(ADHI_gray)
VIJAY_gray=np.dstack(VIJAY_gray)
VISHAL_gray=np.dstack(VISHAL_gray)
```

In [35]:

```
ADHI_gray.shape
```

Out[35]:

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

In [36]:

```
VIJAY_gray.shape
```

Out[36]:

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

In [37]:

```
VISHAL_gray.shape
```

Out[37]:

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

In [38]:

```
ADHI_gray=np.rollaxis(ADHI_gray,axis=2,start=0)
VIJAY_gray=np.rollaxis(VIJAY_gray,axis=2,start=0)
VISHAL_gray=np.rollaxis(VISHAL_gray,axis=2,start=0)
```

In [39]:

```
ADHI_gray.shape
```

Out[39]:

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

In [40]:

```
VIJAY_gray.shape
```

Out[40]:

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

In [41]:

```
VISHAL_gray.shape
```

Out[41]:

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

In [42]:

```
ADHI_gray=ADHI_gray.reshape(len_of_images_ADHI,flatten_size_ADHI)
VIJAY_gray=VIJAY_gray.reshape(len_of_images_VIJAY,flatten_size_VIJAY)
VISHAL_gray=VISHAL_gray.reshape(len_of_images_VISHAL,flatten_size_VISHAL)
```

In [43]:

```
ADHI_gray.shape
```

Out[43]:

```
(10, 262144)
```

In [44]:

```
VIJAY_gray.shape
```

Out[44]:

```
(10, 262144)
```

In [45]:

```
VISHAL_gray.shape
```

Out[45]:

```
(10, 262144)
```

In [46]:

```
ADHI_data=pd.DataFrame(ADHI_gray)
VIJAY_data=pd.DataFrame(VIJAY_gray)
VISHAL_data=pd.DataFrame(VISHAL_gray)
```

In [47]:

```
ADHI_data
```

Out[47]:

	0	1	2	3	4	5	6	7	8	
0	0.844923	0.844923	0.844923	0.844923	0.844923	0.844923	0.844923	0.844923	0.844923	0
1	0.996905	0.996905	0.996905	0.996905	0.996905	0.996905	0.996905	0.996905	0.996905	0
2	0.034958	0.034958	0.034958	0.034958	0.034958	0.033851	0.031868	0.031036	0.031036	0
3	0.588682	0.586936	0.574298	0.564640	0.556214	0.562036	0.570461	0.587173	0.602915	0
4	0.791063	0.791063	0.791063	0.791063	0.791591	0.793016	0.794441	0.795192	0.795527	0
5	0.339397	0.336708	0.324937	0.310859	0.297034	0.292176	0.291691	0.301447	0.314839	0
6	0.992157	0.992157	0.992157	0.992157	0.992157	0.992157	0.992157	0.992157	0.992157	0
7	0.014295	0.014295	0.014295	0.014295	0.014720	0.016121	0.017523	0.018216	0.018216	0
8	0.189288	0.190406	0.189004	0.187603	0.186201	0.184800	0.183398	0.181288	0.178484	0
9	0.969661	0.969661	0.969661	0.969661	0.969661	0.969661	0.969252	0.968747	0.967990	0

10 rows × 262144 columns

In [48]:

```
VIJAY_data
```

Out[48]:

	0	1	2	3	4	5	6	7	8	
0	0.072710	0.072710	0.072710	0.074495	0.076632	0.076632	0.076632	0.076632	0.077604	0
1	0.409148	0.409148	0.409148	0.409148	0.409148	0.409148	0.409148	0.409148	0.409148	0
2	0.959123	0.959123	0.959123	0.960322	0.961769	0.962232	0.962932	0.964404	0.965236	0
3	0.658698	0.658163	0.635041	0.607708	0.576711	0.552243	0.532948	0.521542	0.515825	0
4	0.581161	0.583764	0.593191	0.598159	0.600545	0.593721	0.582653	0.566729	0.553466	0
5	0.984314	0.984314	0.984314	0.984314	0.984314	0.984314	0.984314	0.984314	0.984314	0
6	0.744120	0.750235	0.761133	0.771833	0.777228	0.778739	0.779125	0.775513	0.773036	0
7	0.721822	0.721788	0.722482	0.722356	0.721135	0.718923	0.715093	0.710453	0.704181	0
8	0.995513	0.995513	0.995513	0.995620	0.996861	0.998101	0.998727	0.999210	0.999435	0
9	0.153517	0.153946	0.156121	0.156565	0.156565	0.158725	0.160900	0.163076	0.165251	0

10 rows × 262144 columns

In [49]:

```
VISHAL_data
```

Out[49]:

	0	1	2	3	4	5	6	7	8	
0	0.631671	0.631671	0.631671	0.633105	0.634455	0.635227	0.636396	0.637922	0.638830	0
1	0.809423	0.800196	0.811762	0.823327	0.830778	0.828777	0.826776	0.832890	0.846945	0
2	0.553870	0.554091	0.554861	0.554913	0.555246	0.556767	0.557907	0.558677	0.560043	0
3	0.366523	0.366523	0.366523	0.366523	0.366523	0.366523	0.366523	0.366523	0.366523	0
4	0.362256	0.361781	0.363505	0.365302	0.367882	0.370462	0.374426	0.378712	0.384199	0
5	0.180224	0.180270	0.182254	0.185222	0.189047	0.192973	0.196981	0.202776	0.209861	0
6	0.515637	0.515584	0.513314	0.514605	0.518995	0.530133	0.546619	0.566665	0.589278	0
7	0.240889	0.240889	0.240889	0.240889	0.241321	0.243596	0.244810	0.244810	0.245374	0
8	0.069985	0.069985	0.069985	0.069985	0.069985	0.069985	0.069985	0.069985	0.069985	0
9	0.865950	0.865950	0.865950	0.865950	0.865950	0.865950	0.865950	0.865950	0.865950	0

10 rows × 262144 columns

In [50]:

```
ADHI_data["label"]="ADHI"
VIJAY_data["label"]="VIJAY"
VISHAL_data["label"]="VISHAL"
```

In [51]:

```
ADHI_data
```

Out[51]:

	0	1	2	3	4	5	6	7	8	
0	0.844923	0.844923	0.844923	0.844923	0.844923	0.844923	0.844923	0.844923	0.844923	0
1	0.996905	0.996905	0.996905	0.996905	0.996905	0.996905	0.996905	0.996905	0.996905	0
2	0.034958	0.034958	0.034958	0.034958	0.034958	0.033851	0.031868	0.031036	0.031036	0
3	0.588682	0.586936	0.574298	0.564640	0.556214	0.562036	0.570461	0.587173	0.602915	0
4	0.791063	0.791063	0.791063	0.791063	0.791591	0.793016	0.794441	0.795192	0.795527	0
5	0.339397	0.336708	0.324937	0.310859	0.297034	0.292176	0.291691	0.301447	0.314839	0
6	0.992157	0.992157	0.992157	0.992157	0.992157	0.992157	0.992157	0.992157	0.992157	0
7	0.014295	0.014295	0.014295	0.014295	0.014720	0.016121	0.017523	0.018216	0.018216	0
8	0.189288	0.190406	0.189004	0.187603	0.186201	0.184800	0.183398	0.181288	0.178484	0
9	0.969661	0.969661	0.969661	0.969661	0.969661	0.969661	0.969252	0.968747	0.967990	0

10 rows × 262145 columns

In [52]:

```
VIJAY_data
```

Out[52]:

	0	1	2	3	4	5	6	7	8	
0	0.072710	0.072710	0.072710	0.074495	0.076632	0.076632	0.076632	0.076632	0.077604	0
1	0.409148	0.409148	0.409148	0.409148	0.409148	0.409148	0.409148	0.409148	0.409148	0
2	0.959123	0.959123	0.959123	0.960322	0.961769	0.962232	0.962932	0.964404	0.965236	0
3	0.658698	0.658163	0.635041	0.607708	0.576711	0.552243	0.532948	0.521542	0.515825	0
4	0.581161	0.583764	0.593191	0.598159	0.600545	0.593721	0.582653	0.566729	0.553466	0
5	0.984314	0.984314	0.984314	0.984314	0.984314	0.984314	0.984314	0.984314	0.984314	0
6	0.744120	0.750235	0.761133	0.771833	0.777228	0.778739	0.779125	0.775513	0.773036	0
7	0.721822	0.721788	0.722482	0.722356	0.721135	0.718923	0.715093	0.710453	0.704181	0
8	0.995513	0.995513	0.995513	0.995620	0.996861	0.998101	0.998727	0.999210	0.999435	0
9	0.153517	0.153946	0.156121	0.156565	0.156565	0.158725	0.160900	0.163076	0.165251	0

10 rows × 262145 columns

In [53]:

```
VISHAL_data
```

Out[53]:

	0	1	2	3	4	5	6	7	8	
0	0.631671	0.631671	0.631671	0.633105	0.634455	0.635227	0.636396	0.637922	0.638830	0
1	0.809423	0.800196	0.811762	0.823327	0.830778	0.828777	0.826776	0.832890	0.846945	0
2	0.553870	0.554091	0.554861	0.554913	0.555246	0.556767	0.557907	0.558677	0.560043	0
3	0.366523	0.366523	0.366523	0.366523	0.366523	0.366523	0.366523	0.366523	0.366523	0
4	0.362256	0.361781	0.363505	0.365302	0.367882	0.370462	0.374426	0.378712	0.384199	0
5	0.180224	0.180270	0.182254	0.185222	0.189047	0.192973	0.196981	0.202776	0.209861	0
6	0.515637	0.515584	0.513314	0.514605	0.518995	0.530133	0.546619	0.566665	0.589278	0
7	0.240889	0.240889	0.240889	0.240889	0.241321	0.243596	0.244810	0.244810	0.245374	0
8	0.069985	0.069985	0.069985	0.069985	0.069985	0.069985	0.069985	0.069985	0.069985	0
9	0.865950	0.865950	0.865950	0.865950	0.865950	0.865950	0.865950	0.865950	0.865950	0

10 rows × 262145 columns

In [54]:

```
actor_1=pd.concat([ADHI_data,VIJAY_data])
actor=pd.concat([actor_1,VISHAL_data])
actor
```

Out[54]:

	0	1	2	3	4	5	6	7	8	
0	0.844923	0.844923	0.844923	0.844923	0.844923	0.844923	0.844923	0.844923	0.844923	0
1	0.996905	0.996905	0.996905	0.996905	0.996905	0.996905	0.996905	0.996905	0.996905	0
2	0.034958	0.034958	0.034958	0.034958	0.034958	0.033851	0.031868	0.031036	0.031036	0
3	0.588682	0.586936	0.574298	0.564640	0.556214	0.562036	0.570461	0.587173	0.602915	0
4	0.791063	0.791063	0.791063	0.791063	0.791591	0.793016	0.794441	0.795192	0.795527	0
5	0.339397	0.336708	0.324937	0.310859	0.297034	0.292176	0.291691	0.301447	0.314839	0
6	0.992157	0.992157	0.992157	0.992157	0.992157	0.992157	0.992157	0.992157	0.992157	0
7	0.014295	0.014295	0.014295	0.014295	0.014720	0.016121	0.017523	0.018216	0.018216	0
8	0.189288	0.190406	0.189004	0.187603	0.186201	0.184800	0.183398	0.181288	0.178484	0
9	0.969661	0.969661	0.969661	0.969661	0.969661	0.969661	0.969252	0.968747	0.967990	0
0	0.072710	0.072710	0.072710	0.074495	0.076632	0.076632	0.076632	0.076632	0.077604	0
1	0.409148	0.409148	0.409148	0.409148	0.409148	0.409148	0.409148	0.409148	0.409148	0
2	0.959123	0.959123	0.959123	0.960322	0.961769	0.962232	0.962932	0.964404	0.965236	0
3	0.658698	0.658163	0.635041	0.607708	0.576711	0.552243	0.532948	0.521542	0.515825	0
4	0.581161	0.583764	0.593191	0.598159	0.600545	0.593721	0.582653	0.566729	0.553466	0
5	0.984314	0.984314	0.984314	0.984314	0.984314	0.984314	0.984314	0.984314	0.984314	0
6	0.744120	0.750235	0.761133	0.771833	0.777228	0.778739	0.779125	0.775513	0.773036	0
7	0.721822	0.721788	0.722482	0.722356	0.721135	0.718923	0.715093	0.710453	0.704181	0
8	0.995513	0.995513	0.995513	0.995620	0.996861	0.998101	0.998727	0.999210	0.999435	0
9	0.153517	0.153946	0.156121	0.156565	0.156565	0.158725	0.160900	0.163076	0.165251	0
0	0.631671	0.631671	0.631671	0.633105	0.634455	0.635227	0.636396	0.637922	0.638830	0
1	0.809423	0.800196	0.811762	0.823327	0.830778	0.828777	0.826776	0.832890	0.846945	0
2	0.553870	0.554091	0.554861	0.554913	0.555246	0.556767	0.557907	0.558677	0.560043	0
3	0.366523	0.366523	0.366523	0.366523	0.366523	0.366523	0.366523	0.366523	0.366523	0
4	0.362256	0.361781	0.363505	0.365302	0.367882	0.370462	0.374426	0.378712	0.384199	0
5	0.180224	0.180270	0.182254	0.185222	0.189047	0.192973	0.196981	0.202776	0.209861	0
6	0.515637	0.515584	0.513314	0.514605	0.518995	0.530133	0.546619	0.566665	0.589278	0
7	0.240889	0.240889	0.240889	0.240889	0.241321	0.243596	0.244810	0.244810	0.245374	0
8	0.069985	0.069985	0.069985	0.069985	0.069985	0.069985	0.069985	0.069985	0.069985	0
9	0.865950	0.865950	0.865950	0.865950	0.865950	0.865950	0.865950	0.865950	0.865950	0

30 rows × 262145 columns

In [55]:

```
from sklearn.utils import shuffle
kollywood_indexed=shuffle(actor).reset_index()
kollywood_indexed
```

Out[55]:

	index	0	1	2	3	4	5	6	7	
0	9	0.969661	0.969661	0.969661	0.969661	0.969661	0.969661	0.969252	0.968747	0.9
1	8	0.069985	0.069985	0.069985	0.069985	0.069985	0.069985	0.069985	0.069985	0.0
2	6	0.515637	0.515584	0.513314	0.514605	0.518995	0.530133	0.546619	0.566665	0.5
3	9	0.865950	0.865950	0.865950	0.865950	0.865950	0.865950	0.865950	0.865950	0.8
4	0	0.072710	0.072710	0.072710	0.074495	0.076632	0.076632	0.076632	0.076632	0.0
5	6	0.992157	0.992157	0.992157	0.992157	0.992157	0.992157	0.992157	0.992157	0.9
6	3	0.366523	0.366523	0.366523	0.366523	0.366523	0.366523	0.366523	0.366523	0.3
7	2	0.553870	0.554091	0.554861	0.554913	0.555246	0.556767	0.557907	0.558677	0.5
8	7	0.240889	0.240889	0.240889	0.240889	0.241321	0.243596	0.244810	0.244810	0.2
9	9	0.153517	0.153946	0.156121	0.156565	0.156565	0.158725	0.160900	0.163076	0.1
10	5	0.339397	0.336708	0.324937	0.310859	0.297034	0.292176	0.291691	0.301447	0.3
11	0	0.631671	0.631671	0.631671	0.633105	0.634455	0.635227	0.636396	0.637922	0.6
12	5	0.180224	0.180270	0.182254	0.185222	0.189047	0.192973	0.196981	0.202776	0.2
13	2	0.034958	0.034958	0.034958	0.034958	0.034958	0.033851	0.031868	0.031036	0.0
14	7	0.721822	0.721788	0.722482	0.722356	0.721135	0.718923	0.715093	0.710453	0.7
15	5	0.984314	0.984314	0.984314	0.984314	0.984314	0.984314	0.984314	0.984314	0.9
16	3	0.588682	0.586936	0.574298	0.564640	0.556214	0.562036	0.570461	0.587173	0.6
17	1	0.996905	0.996905	0.996905	0.996905	0.996905	0.996905	0.996905	0.996905	0.9
18	8	0.995513	0.995513	0.995513	0.995620	0.996861	0.998101	0.998727	0.999210	0.9
19	4	0.581161	0.583764	0.593191	0.598159	0.600545	0.593721	0.582653	0.566729	0.5
20	1	0.409148	0.409148	0.409148	0.409148	0.409148	0.409148	0.409148	0.409148	0.4
21	8	0.189288	0.190406	0.189004	0.187603	0.186201	0.184800	0.183398	0.181288	0.1
22	6	0.744120	0.750235	0.761133	0.771833	0.777228	0.778739	0.779125	0.775513	0.7
23	4	0.362256	0.361781	0.363505	0.365302	0.367882	0.370462	0.374426	0.378712	0.3
24	7	0.014295	0.014295	0.014295	0.014295	0.014720	0.016121	0.017523	0.018216	0.0
25	1	0.809423	0.800196	0.811762	0.823327	0.830778	0.828777	0.826776	0.832890	0.8
26	2	0.959123	0.959123	0.959123	0.960322	0.961769	0.962232	0.962932	0.964404	0.9
27	0	0.844923	0.844923	0.844923	0.844923	0.844923	0.844923	0.844923	0.844923	0.8
28	3	0.658698	0.658163	0.635041	0.607708	0.576711	0.552243	0.532948	0.521542	0.5
29	4	0.791063	0.791063	0.791063	0.791063	0.791591	0.793016	0.794441	0.795192	0.7

30 rows × 262146 columns

In [56]:

```
kollywood_actors=kollywood_indexed.drop(['index'],axis=1)
kollywood_actors
```

Out[56]:

	0	1	2	3	4	5	6	7	8
0	0.969661	0.969661	0.969661	0.969661	0.969661	0.969661	0.969252	0.968747	0.967990
1	0.069985	0.069985	0.069985	0.069985	0.069985	0.069985	0.069985	0.069985	0.069985
2	0.515637	0.515584	0.513314	0.514605	0.518995	0.530133	0.546619	0.566665	0.589278
3	0.865950	0.865950	0.865950	0.865950	0.865950	0.865950	0.865950	0.865950	0.865950
4	0.072710	0.072710	0.072710	0.074495	0.076632	0.076632	0.076632	0.076632	0.077604
5	0.992157	0.992157	0.992157	0.992157	0.992157	0.992157	0.992157	0.992157	0.992157
6	0.366523	0.366523	0.366523	0.366523	0.366523	0.366523	0.366523	0.366523	0.366523
7	0.553870	0.554091	0.554861	0.554913	0.555246	0.556767	0.557907	0.558677	0.560043
8	0.240889	0.240889	0.240889	0.240889	0.241321	0.243596	0.244810	0.244810	0.245374
9	0.153517	0.153946	0.156121	0.156565	0.156565	0.158725	0.160900	0.163076	0.165251
10	0.339397	0.336708	0.324937	0.310859	0.297034	0.292176	0.291691	0.301447	0.314839
11	0.631671	0.631671	0.631671	0.633105	0.634455	0.635227	0.636396	0.637922	0.638830
12	0.180224	0.180270	0.182254	0.185222	0.189047	0.192973	0.196981	0.202776	0.209861
13	0.034958	0.034958	0.034958	0.034958	0.034958	0.033851	0.031868	0.031036	0.031036
14	0.721822	0.721788	0.722482	0.722356	0.721135	0.718923	0.715093	0.710453	0.704181
15	0.984314	0.984314	0.984314	0.984314	0.984314	0.984314	0.984314	0.984314	0.984314
16	0.588682	0.586936	0.574298	0.564640	0.556214	0.562036	0.570461	0.587173	0.602915
17	0.996905	0.996905	0.996905	0.996905	0.996905	0.996905	0.996905	0.996905	0.996905
18	0.995513	0.995513	0.995513	0.995620	0.996861	0.998101	0.998727	0.999210	0.999435
19	0.581161	0.583764	0.593191	0.598159	0.600545	0.593721	0.582653	0.566729	0.553466
20	0.409148	0.409148	0.409148	0.409148	0.409148	0.409148	0.409148	0.409148	0.409148
21	0.189288	0.190406	0.189004	0.187603	0.186201	0.184800	0.183398	0.181288	0.178484
22	0.744120	0.750235	0.761133	0.771833	0.777228	0.778739	0.779125	0.775513	0.773036
23	0.362256	0.361781	0.363505	0.365302	0.367882	0.370462	0.374426	0.378712	0.384199
24	0.014295	0.014295	0.014295	0.014295	0.014720	0.016121	0.017523	0.018216	0.018216
25	0.809423	0.800196	0.811762	0.823327	0.830778	0.828777	0.826776	0.832890	0.846945
26	0.959123	0.959123	0.959123	0.960322	0.961769	0.962232	0.962932	0.964404	0.965236
27	0.844923	0.844923	0.844923	0.844923	0.844923	0.844923	0.844923	0.844923	0.844923
28	0.658698	0.658163	0.635041	0.607708	0.576711	0.552243	0.532948	0.521542	0.515825
29	0.791063	0.791063	0.791063	0.791063	0.791591	0.793016	0.794441	0.795192	0.795527

30 rows × 262145 columns

In [57]:

```
kollywood_actors.to_csv("C:/Users/Magesh Kumar/Documents/IMAGE PROCESSING/ADHI/kollywood_ac
```

In [58]:

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

Out[58]:

```
array([[0.9696611764705884, 0.9696611764705885, 0.9696611764705884, ...,
        0.6199243580130035, 0.5972698777520422, 0.6035124456461738],
       [0.0699850980392157, 0.06998509803921571, 0.06998509803921571,
        ..., 0.14012673331765568, 0.13143431450039733,
        0.1301163304198022],
       [0.5156368711703431, 0.5155842884497549, 0.5133144676776961, ...,
        0.27018998161764707, 0.26423869485294116, 0.26410082720588235],
       ...,
       [0.8449227450980392, 0.8449227450980392, 0.8449227450980392, ...,
        0.02555321548761106, 0.025384122894885494, 0.04399413019516889],
       [0.6586983023041371, 0.6581626727444518, 0.6350413300847072, ...,
        0.4453019607843137, 0.4453019607843138, 0.4453019607843138],
       [0.7910627450980392, 0.7910627450980392, 0.7910627450980392, ...,
        0.0721950919117647, 0.07219509191176471, 0.0721950919117647]],
      dtype=object)
```

In [59]:

```
y
```

Out[59]:

```
array(['ADHI', 'VISHAL', 'VISHAL', 'VISHAL', 'VIJAY', 'ADHI', 'VISHAL',
       'VISHAL', 'VISHAL', 'VIJAY', 'ADHI', 'VISHAL', 'VISHAL', 'ADHI',
       'VIJAY', 'VIJAY', 'ADHI', 'ADHI', 'VIJAY', 'VIJAY', 'VIJAY',
       'ADHI', 'VIJAY', 'VISHAL', 'ADHI', 'VISHAL', 'VIJAY', 'ADHI',
       'VIJAY', 'ADHI'], dtype=object)
```

In [83]:

```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.5,random_state=0)
```

In [84]:

```
from sklearn import svm
```

In [85]:

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

Out[85]:

```
SVC()
```

In [86]:

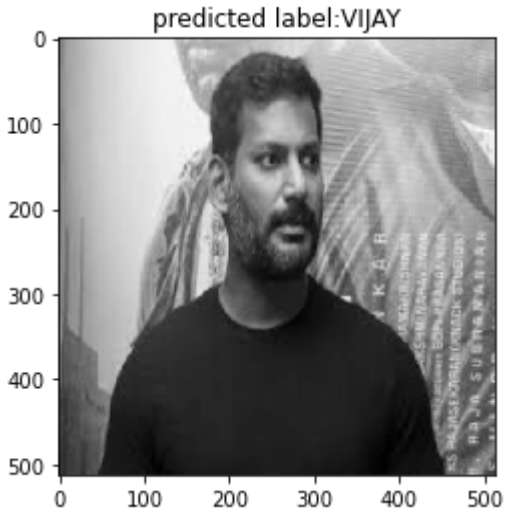
```
y_pred =clf.predict(x_test)
y_pred
```

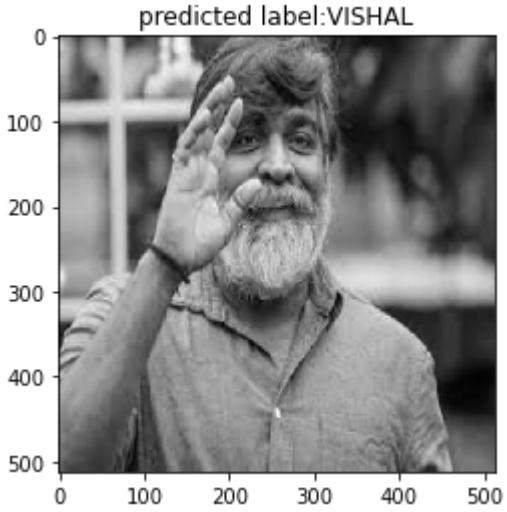
Out[86]:

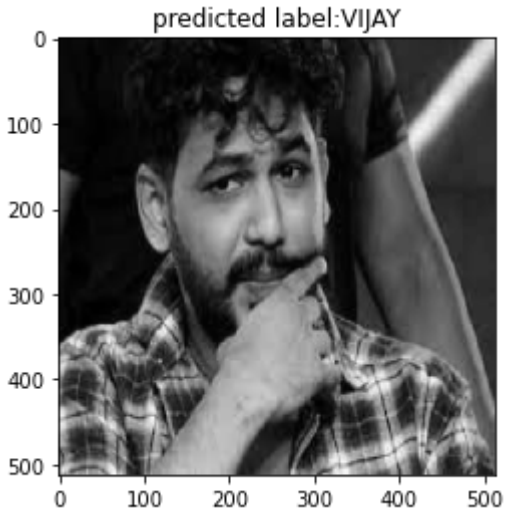
```
array(['VIJAY', 'VISHAL', 'VIJAY', 'VIJAY', 'VISHAL', 'VIJAY', 'VIJAY',
       'VIJAY', 'VIJAY', 'VIJAY', 'VIJAY', 'VIJAY', 'VIJAY', 'VIJAY',
       'VISHAL'], dtype=object)
```

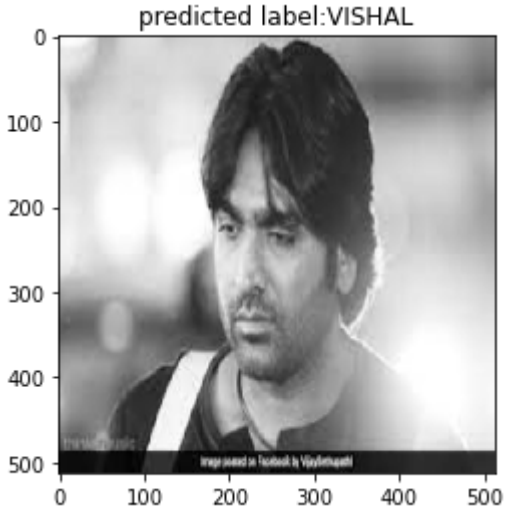
In [87]:

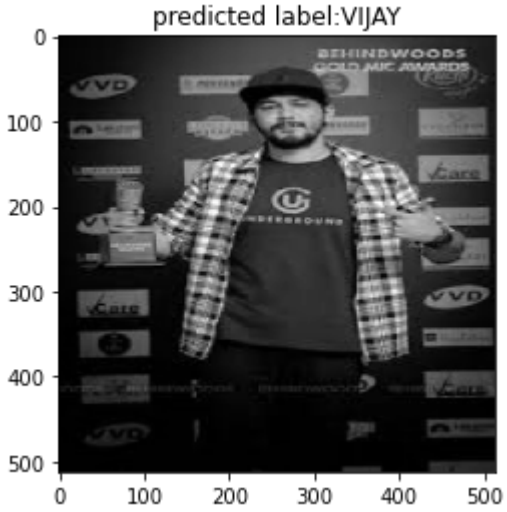
```
for i in range (6):
    predicted_images = (np.reshape(x_test[i],(512,512)).astype(np.float64))
    plt.title('predicted label:{0}'.format(y_pred[i]))
    plt.imshow(predicted_images,interpolation='nearest',cmap='gray')
    plt.show()
```











In [71]:

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

Out[71]:

```
0.08333333333333333
```

In [72]:

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

Out[72]:

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

In [ ]:

In [ ]: