Computer Vision

> it is a field of AI that perform prediction on images and videos. there are so many libs to work with computer vision: opencv dlib google vision mediapipe etc.

In [1]:
```
import cv2
```

In [5]:
```
img1=cv2.imread("f:/images/sachin.jpg",cv2.IMREAD_COLOR)
print(type(img1),img1.ndim)

img2=cv2.imread("f:/images/sachin.jpg",cv2.IMREAD_GRAYSCALE)
print(type(img2),img2.ndim)
```

```
<class 'numpy.ndarray'> 3
<class 'numpy.ndarray'> 2
```

In [2]:
```
cv2.IMREAD_COLOR
```

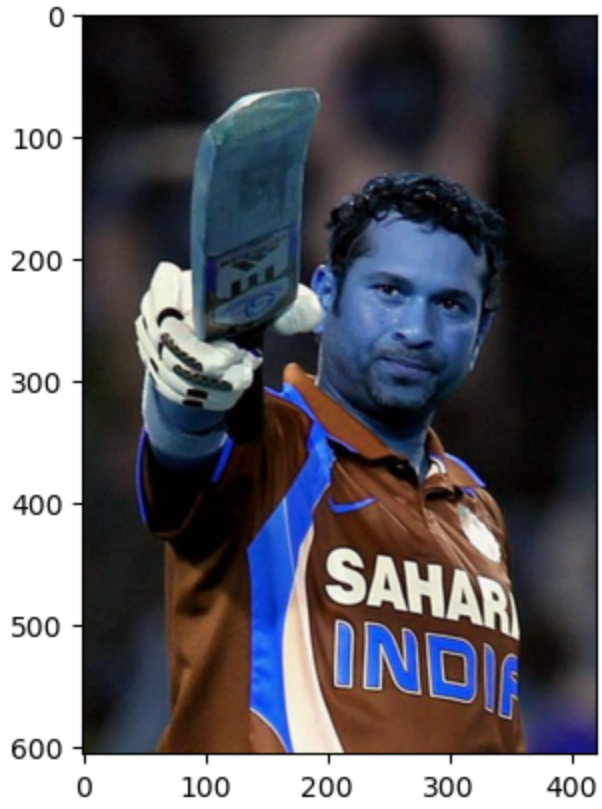
Out[2]:
```
1
```

In [3]:
```
cv2.IMREAD_GRAYSCALE
```

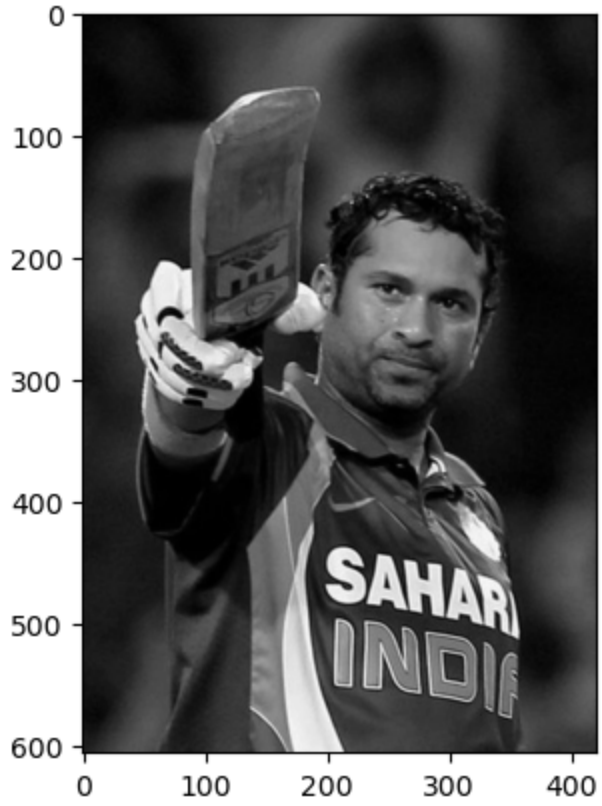
Out[3]:
```
0
```

In [6]:
```
import matplotlib.pyplot as plt
```

In [12]:
```
plt.imshow(img1,cmap="winter")
plt.show()
```



In [13]:
```
plt.imshow(img2,cmap="gray")
plt.show()
```



In [17]:
```
cv2.imshow("clr_img",img1)     #display img on separate window
cv2.waitKey()                  #hold window until user press any key
cv2.destroyWindow("clr_img")   #to free window from memory
```

In [21]:
```
cv2.namedWindow("clr_win",cv2.WINDOW_NORMAL)   #to make resizable window
cv2.imshow("clr_win",img1)                     #display img on separate window
cv2.waitKey()                                  #hold window until user press any key
cv2.destroyWindow("clr_win")                   #to free window from memory
```

In [22]:
```
cv2.imshow("clrimg",img1)
cv2.imshow("grayimg",img2)
cv2.waitKey()
cv2.destroyAllWindows()
```

In [23]:
```
cv2.imshow("clrimg",img1)
cv2.waitKey(5000)    #timeout in millis
cv2.destroyAllWindows()
```

In [26]:
```
print(type(img1))
print(img1.ndim)
print(img1.shape) #h,w,color_channels
```

```
<class 'numpy.ndarray'>
3
(606, 420, 3)
```

In [27]:
```
img1
```

Out[27]:
```
array([[[ 11,  12,  10],
        [ 11,  12,  10],
        [ 10,  11,   9],
        ...,
        [ 22,  23,  33],
        [ 22,  24,  34],
        [ 22,  24,  34]],

       [[  7,   8,   6],
```

```
 [  8,   9,   7],
 [  9,  10,   8],
 ...,
 [ 23,  24,  34],
 [ 23,  25,  35],
 [ 23,  25,  35]],

[[  4,   5,   3],
 [  6,   7,   5],
 [  8,   9,   7],
 ...,
 [ 25,  26,  36],
 [ 25,  27,  37],
 [ 25,  27,  37]],

...,

[[ 30,  32,  12],
 [ 33,  35,  15],
 [ 37,  38,  18],
 ...,
 [ 26,  27, 115],
 [ 28,  26, 115],
 [ 27,  25, 114]],

[[ 35,  37,  17],
 [ 39,  40,  20],
 [ 41,  42,  22],
 ...,
 [ 28,  28, 118],
 [ 28,  28, 118],
 [ 29,  27, 117]],

[[ 39,  41,  21],
 [ 42,  43,  23],
 [ 44,  45,  25],
 ...,
 [ 31,  31, 121],
 [ 31,  31, 121],
 [ 34,  32, 122]]], dtype=uint8)
```

In [ ]: