☑ **Yating Liao**

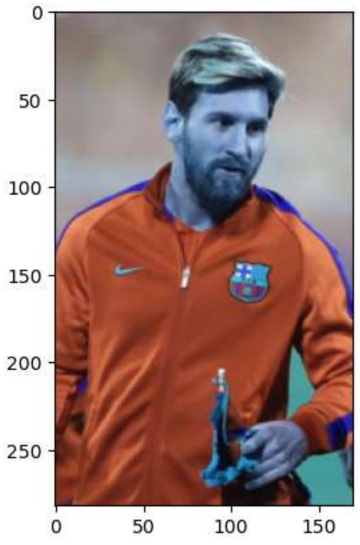
Homework #3 ISE 599 Deep Learning Student ID: 7636428840

1. (10 pts.) Open the library and get the player image

```
In [1]:
import cv2
import pandas as pd
import matplotlib.pyplot as plt

image= cv2.imread('D:\Caroline\Documents\Graduate\ISE 599  Deep Learning\player1.png')
plt.imshow(image)
```

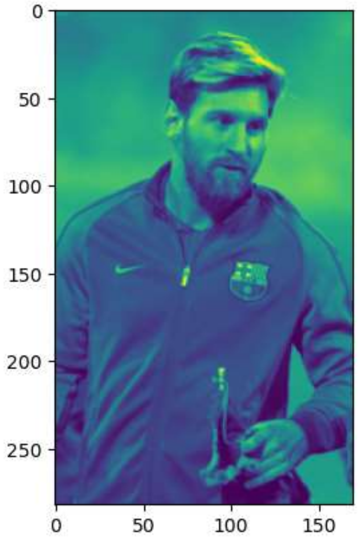
```
Out[1]: <matplotlib.image.AxesImage at 0x254155ba3e0>
```



2. (10 pts.) Use the cvtColor function to convert the image to grayscale, then display it

```
In [2]:
image_gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
plt.imshow(image_gray)
```

```
Out[2]: <matplotlib.image.AxesImage at 0x25415eda5c0>
```



3. (10 pts.) Load the Haar cascade face classifier xml file.

```
In [3]:  haar_cascade = cv2.CascadeClassifier('D:\Caroline\Documents\Graduate\ISE 599  Deep Learning\haarcascade_frontalface_default.xml')
```

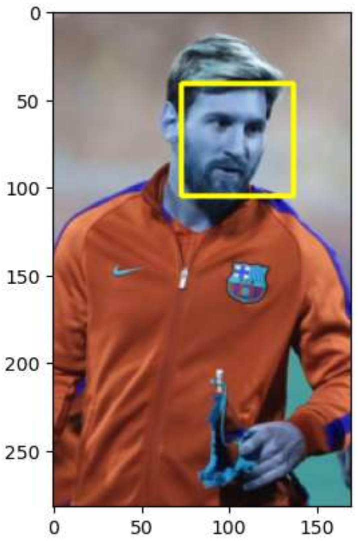
4. (10 pts.) Get coordinates for the player face in the image

```
In [4]:  faces_cordinates = haar_cascade.detectMultiScale(image_gray,scaleFactor = 1.3,minNeighbors = 7)
```

5. (10 pts.) Use the rectangle function to draw a rectangle on the face

```
In [5]:  for (p,q,r,s) in faces_cordinates:
             cv2.rectangle(image, (p, q), (p+r, q+s), (255,255,0), 2)
         plt.imshow(image)

Out[5]:  <matplotlib.image.AxesImage at 0x25415f66ad0>
```



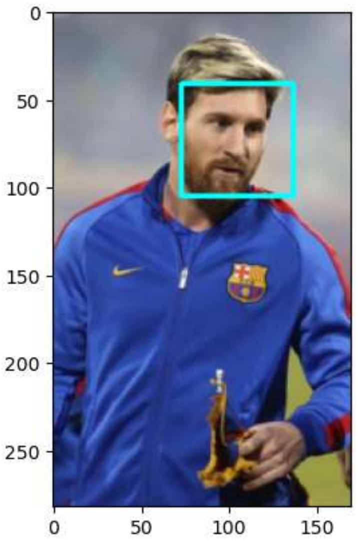
6. (10 pts.) Convert the image color space BGR to RGB

```
In [6]:  image_rgb = cv2.cvtColor(image,cv2.COLOR_BGR2RGB)
```

7. (10 pts.) Display the image with the bounding box

```
In [7]:  plt.imshow(image_rgb)

Out[7]:  <matplotlib.image.AxesImage at 0x25416faa4d0>
```



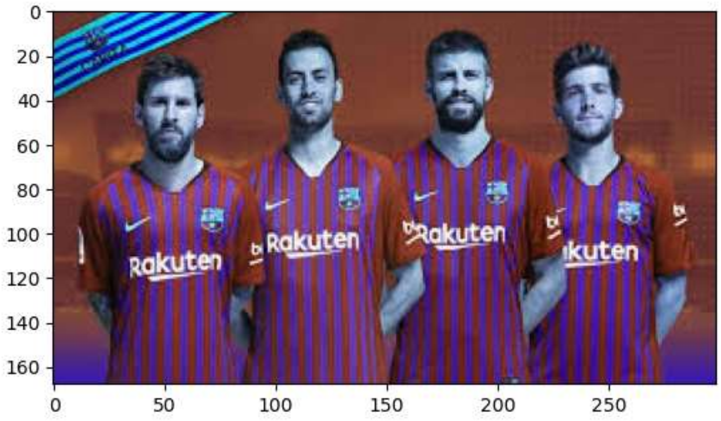
8. (30 pts.) Repeat these steps to detect the faces on the file players.png

In [8]:

```
# get the players image
image2 = cv2.imread("D:\Caroline\Documents\Graduate\ISE 599  Deep Learning\players.jpeg")
plt.imshow(image2)
```

Out[8]:

```
<matplotlib.image.AxesImage at 0x2541701a0e0>
```

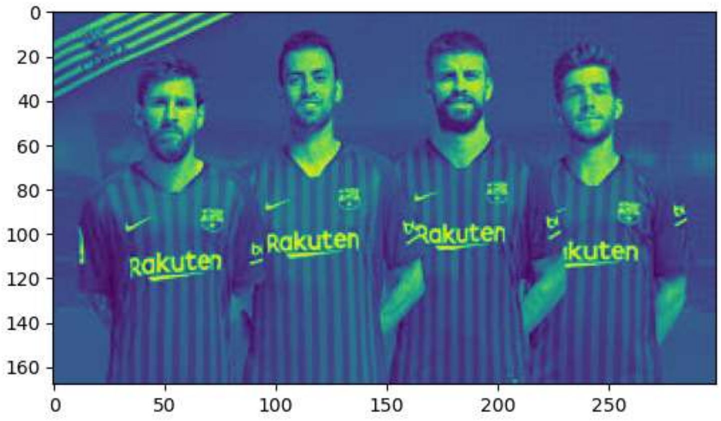


In [9]:

```
# Use the cvtColor function to convert the image to grayscale, then display it
image_gray2 = cv2.cvtColor(image2, cv2.COLOR_BGR2GRAY)
plt.imshow(image_gray2)
```

Out[9]:

```
<matplotlib.image.AxesImage at 0x25419c41270>
```



In [10]:

```
# Get coordinates for the player face in the image
faces_cordinates2 = haar_cascade.detectMultiScale(image_gray2)
```

In [11]:

```
# Use the rectangle function to draw a rectangle on the face
for (p,q,r,s) in faces_cordinates2:
    cv2.rectangle(image2, (p, q), (p+r, q+s), (255,255,0), 2)
```

In [12]:

```
# Convert the image color space BGR to RGB
image_rgb2 = cv2.cvtColor(image2, cv2.COLOR_BGR2RGB)
```

In [13]:

```
plt.imshow(image_rgb2)
```

Out[13]:

```
<matplotlib.image.AxesImage at 0x25419cd0100>
```