#https://www.kaggle.com/serkanpeldek/face-detection-with-opencv#
(https://www.kaggle.com/serkanpeldek/face-detection-with-opencv)

In [1]:
```
# pip install opencv-python
```

In [2]:
```
import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)

#Visiulazation
import matplotlib.pyplot as plt

#image processing
import cv2

#extracting zippped file
import tarfile

#systems
import os
```

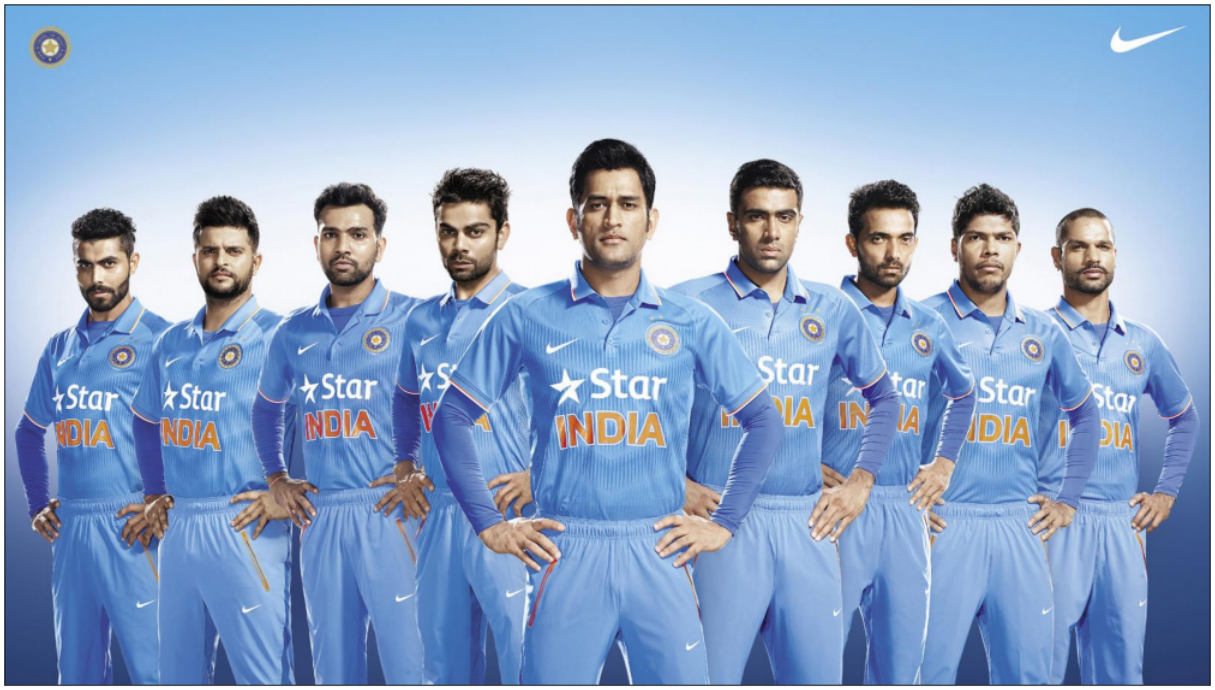
In [3]:
```
# The size by which the shape is enlarged or reduced is called as its scale facto
class FaceDetector():

    def __init__(self,faceCascadePath):
        self.faceCascade=cv2.CascadeClassifier(faceCascadePath)


    def detect(self, image, scaleFactor=1.1,
               minNeighbors=5,
               minSize=(30,30)):

        #function return rectangle coordinates of faces for given image
        rects=self.faceCascade.detectMultiScale(image,
                                                scaleFactor=scaleFactor,
                                                minNeighbors=minNeighbors,
                                                minSize=minSize)
        return rects
```

In [4]:
```
#Frontal face of haar cascade loaded
frontal_cascade_path="/home/hduser/jupyter/Face_Detection_with_OpenCV/haarcascade

#Detector object Macreated
fd=FaceDetector(frontal_cascade_path)
```

In [5]:
```
#An image contains faces, loaded
national_team_org=cv2.imread("/home/hduser/jupyter/Face_Detection_with_OpenCV/b97
```

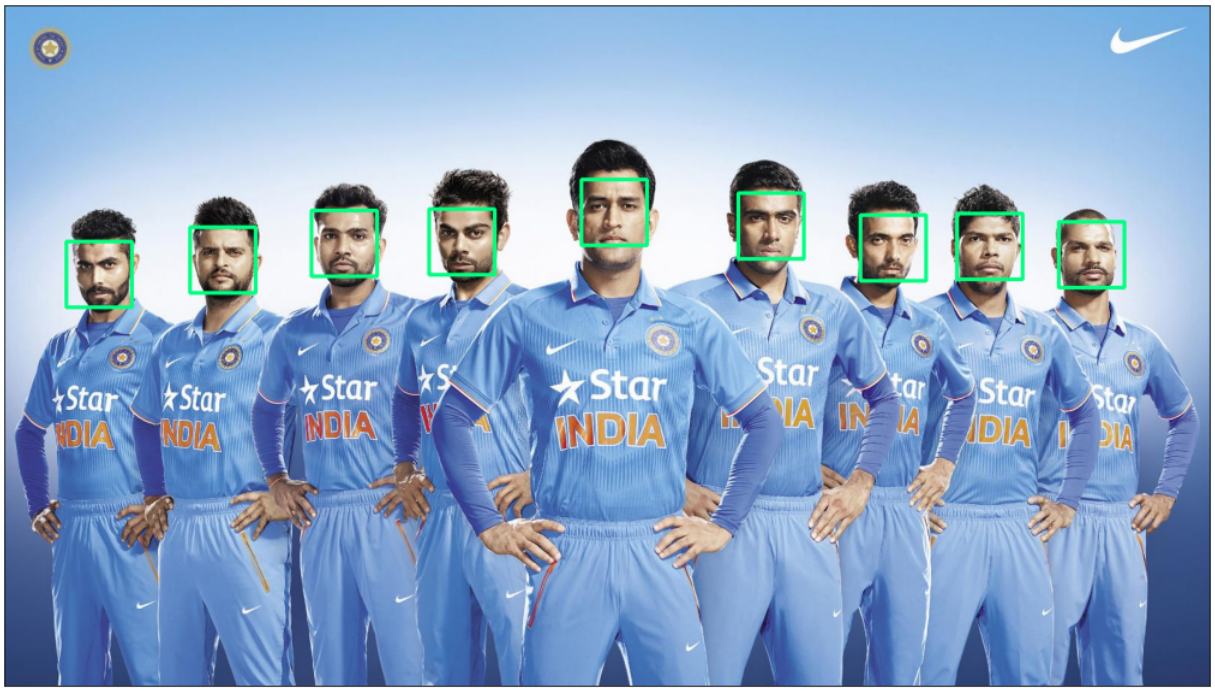
In [6]:
```
def get_national_team():
    return np.copy(national_team_org)

def show_image(image):
    plt.figure(figsize=(18,15))
    #Before showing image, bgr color order transformed to rgb order
    plt.imshow(cv2.cvtColor(image, cv2.COLOR_BGR2RGB))
    plt.xticks([])
    plt.yticks([])
    plt.show()
```

In [7]:
```
show_image(get_national_team())
```

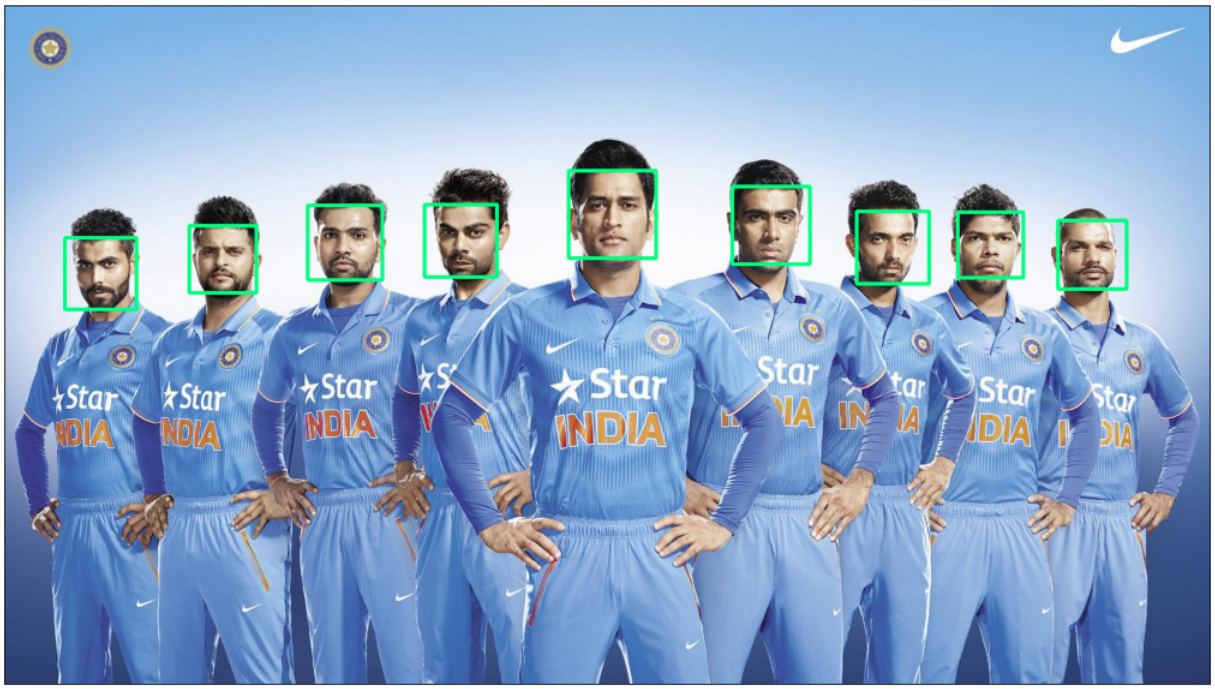


In [8]:
```
def detect_face(image, scaleFactor, minNeighbors, minSize):
    # face will detected in gray image
    image_gray=cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)

    faces=fd.detect(image_gray,
                    scaleFactor=scaleFactor,
                    minNeighbors=minNeighbors,
                    minSize=minSize)

    for x, y, w, h in faces:
        #detected faces shown in color image
        cv2.rectangle(image,(x,y),(x+w, y+h),(127, 255,0),3)

    show_image(image)
```

In [9]:
```
national_team=get_national_team()

detect_face(image=national_team,
            scaleFactor=1.9,
            minNeighbors=3,
            minSize=(30,30))
```



In [10]:
```
national_team=get_national_team()
#Let's play around function parameters
detect_face(image=national_team,
            scaleFactor=1.3,
            minNeighbors=3,
            minSize=(30,30))
```



In [ ]: