Face & Eye Detection using HAAR Cascade Classifiers

1. Importing Necessary Libraries

In [25]:
```
import cv2  # OpenCV library for computer vision tasks
import numpy as np
```

2. Load HAAR Cascade Classifiers

Load the pre-trained HAAR Cascade classifiers for face and eye detection. OpenCV provides these XML files.

In [35]:
```
face_classifier = cv2.CascadeClassifier('C:/projects/facedetect/Haarcascades/haarcascade_frontalface_default.xml')
eye_classifier = cv2.CascadeClassifier('C:/projects/facedetect/Haarcascades/haarcascade_eye.xml')

# Check if classifiers are loaded properly
if face_classifier.empty():
    raise IOError("Failed to load face classifier xml file. Check the path.")

if eye_classifier.empty():
    raise IOError("Failed to load eye classifier xml file. Check the path.")
```

3. Load the image, resize and convert it to grayscale.

In [49]:
```
# Read the input image
image_path = 'C:/projects/facedetect/pic1.jpg'
image = cv2.imread(image_path)
if image is None:
    raise FileNotFoundError("Image file not found. Please check the path.")

# Resize the image for easier display
scale_percent = 50  # percent of original size
width = int(image.shape[1] * scale_percent / 100)
height = int(image.shape[0] * scale_percent / 100)
```

```
dim = (width, height)
resized_image = cv2.resize(image, dim, interpolation=cv2.INTER_AREA)
gray = cv2.cvtColor(resized_image, cv2.COLOR_BGR2GRAY)
```

4. Detect faces in the grayscale image

In [50]:
```
# Detect faces in the grayscale image
faces = face_classifier.detectMultiScale(gray, scaleFactor=1.1, minNeighbors=5, minSize=(30, 30))

# If no faces are detected, face_classifier returns an empty list
if len(faces) == 0:
    print("No faces found")
else:
    # Iterate through faces array and draw a rectangle over each face
    for (x, y, w, h) in faces:
        cv2.rectangle(resized_image, (x, y), (x + w, y + h), (127, 0, 255), 2)
        roi_gray = gray[y:y + h, x:x + w]
        roi_color = resized_image[y:y + h, x:x + w]

        # Detect eyes within the detected face region
        eyes = eye_classifier.detectMultiScale(roi_gray, scaleFactor=1.1, minNeighbors=5, minSize=(20, 20))
        for (ex, ey, ew, eh) in eyes:
            cv2.rectangle(roi_color, (ex, ey), (ex + ew, ey + eh), (0, 255, 0), 2)
```

5. Display result

In [52]:
```
# Display the output image with detected faces and eyes
while True:
    cv2.imshow('Face and Eye Detection', resized_image)
    if cv2.waitKey(1) & 0xFF == ord('q'):  # Press 'q' to quit the display window
        break

cv2.destroyAllWindows()
```

In [ ]: