Face Detection using Haar Cascade Algorithm

Introdduction to Haar Cascade Algorithm:

The Haar Cascade Algorithm is a **machine learning-based method** for detecting objects, especially **faces**, in images or videos.
It uses a **cascade of classifiers trained on Haar-like features** to identify face regions by distinguishing them from non-face areas.

Working :

1. **Haar-like Features**
    - Simple rectangular features capturing intensity differences (like dark eyes vs bright cheeks).
    - Computed efficiently using **integral images** for fast pixel sum calculations.
2. **Cascade Classifier**
    - Trained using **AdaBoost** with positive (face) and negative (non-face) images.
    - Quickly rejects non-face regions early, improving speed.
3. **Sliding Window**
    - A window **scans the image at multiple scales**, checking for faces at each size.
    - Passing windows are marked as face regions.
4. **Post-processing**
    - Overlapping detections are merged to finalize the face locations.

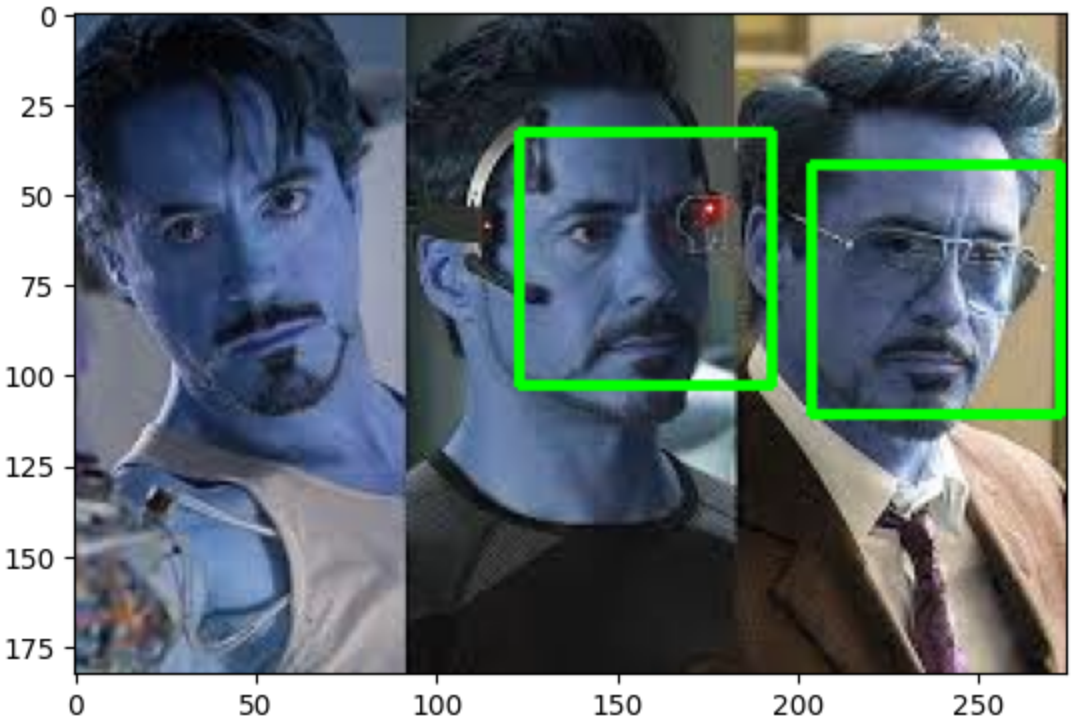
In Python, OpenCV's `cv2.CascadeClassifier` is used with **pre-trained XML files** (like `haarcascade_frontalface_default.xml`).

The `detectMultiScale()` function performs the detection on images or video frames.

```
import cv2 as cv
import numpy as np
img = cv.imread("/content/tonystark.jpg", 1)
gray = cv.cvtColor(img, cv.COLOR_BGR2GRAY)
cascade_dir = "/content/haarcascade_frontalface_default.xml"
face_cascade =cv.CascadeClassifier(cascade_dir)
faces = face_cascade.detectMultiScale(gray, 1.1, 5)
```

```
import matplotlib.pyplot as plt
for (x, y, w, h) in faces:
    img = cv.rectangle (img, (x,y), (x+w, y+h), (0,255,0), 2) # (x,y) - topmost left coordir
plt.imshow(img)
```

```
<matplotlib.image.AxesImage at 0x796c2beeca50>
```



Eye Detection using Haar Cascade Algorithm

```
import cv2 as cv
import numpy as np
img = cv.imread("/content/tonystark.jpg", 1)
gray = cv.cvtColor(img, cv.COLOR_BGR2GRAY)
cascade_dir = "/content/haarcascade_eye.xml"
face_cascade =cv.CascadeClassifier(cascade_dir)
faces = face_cascade.detectMultiScale(gray, 1.1, 5)
```

```
import matplotlib.pyplot as plt
for (x, y, w, h) in faces:
    img = cv.rectangle (img, (x,y), (x+w, y+h), (0,255,0), 2) # (x,y) - topmost left coordir
plt.imshow(img)
```

<matplotlib.image.AxesImage at 0x796c2e15da10>