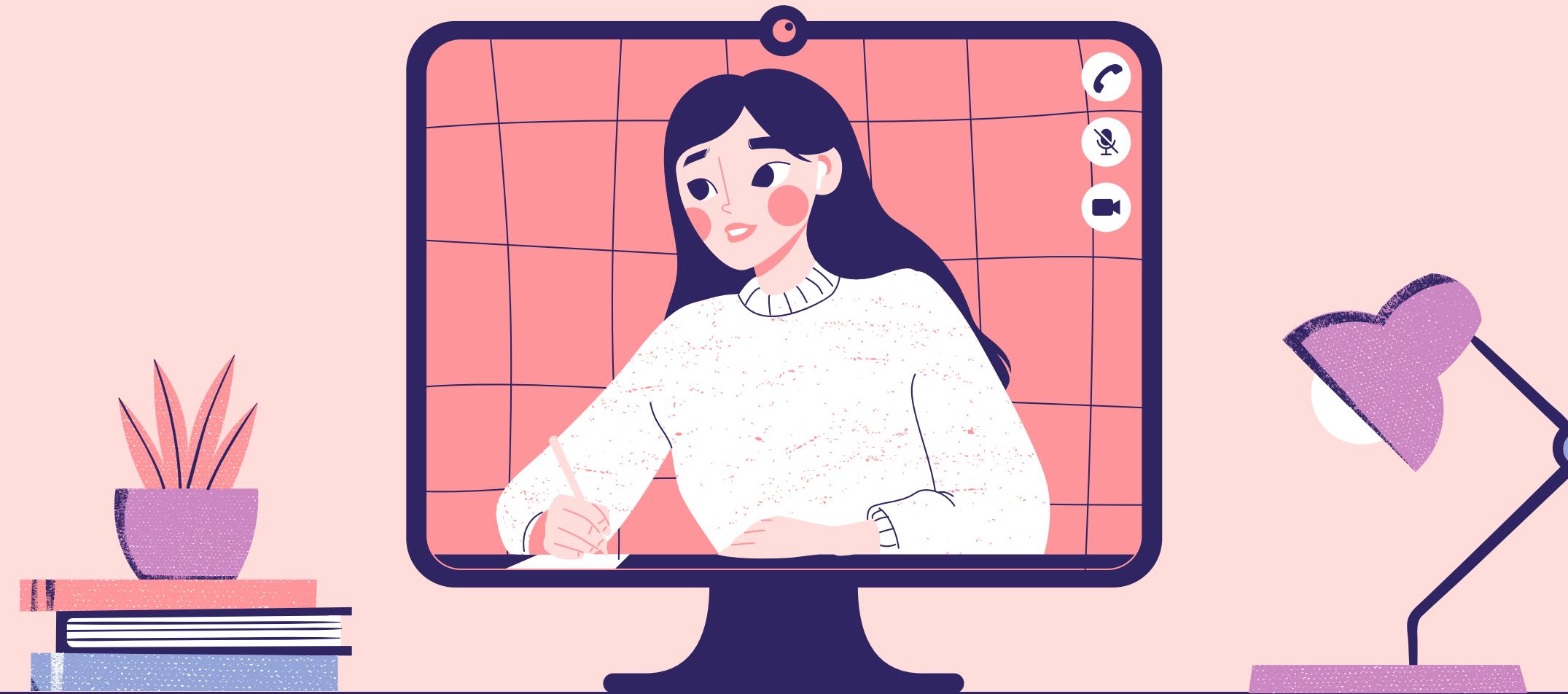


Group Project : by Ninjas

# Gender Detection Model



Using OpenCV model to detect gender

# Agenda

- What is Face Detection?
- What is Gender Detection?
- Our Gender Detection Model
- Real-world Gender Detection

Applications



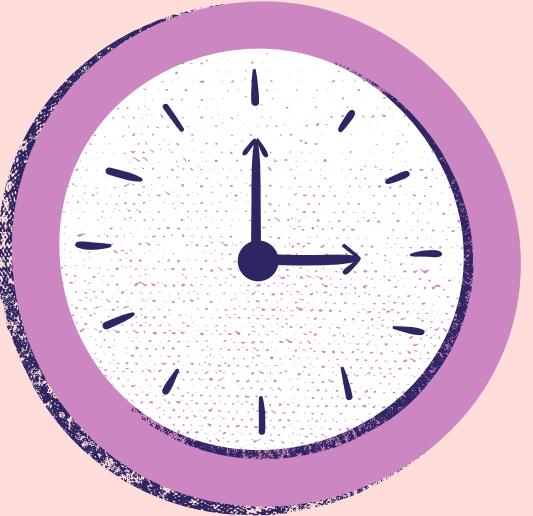
# Introduction



Face detection using Haar cascades is a machine learning based approach where a cascade function is trained with a set of input data. OpenCV already contains many pre-trained classifiers for face, eyes, smiles

# What is Face Detection?

Face detection is a computer technology that determines the location and size of a human face in digital images. Given an image, the goal of facial recognition is to determine whether there are any faces and return the bounding box of each detected face



**Automatic prediction of gender from face images has drawn a lot of attention recently, due to its wide application in various facial analysis problems. However, due to the large variations of face images (such as variation in lighting, scale, and occlusion) the existing models are still behind the desired accuracy level which is necessary for exploiting these models in real-world applications.**

# **Gender detection model**



# Our Gender Detection Model

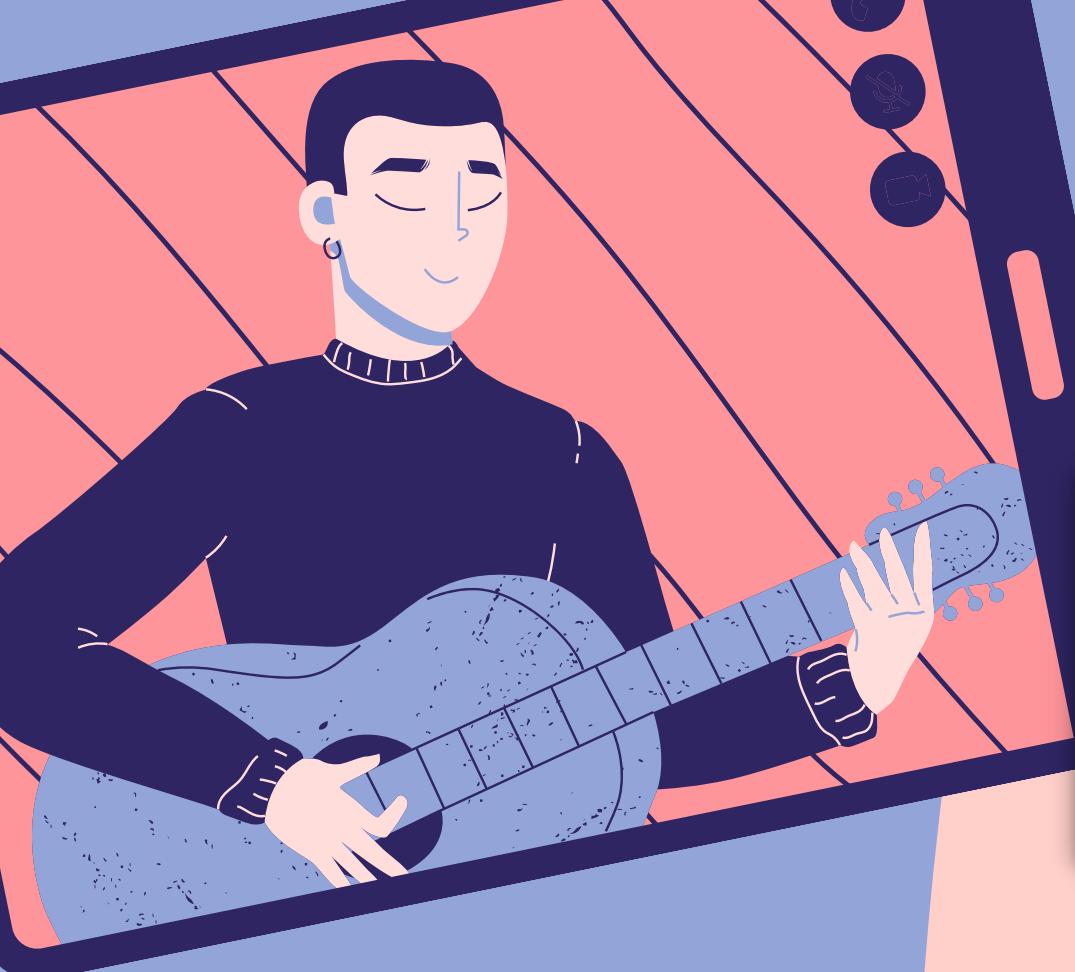
**gender.prototxt**

**The model architecture for the gender  
detection model**

**gender.caffemodel**

**The pre-trained model weights for  
gender detection.**

# Gender detection model



## detect faces

```
def detect_faces(img):
    #Transform the image to grey to detect faces correctly
    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    #detectMultiScale takes three arguments :the input image, scaleFactor and minNeighbours
    faces = haar_detector.detectMultiScale(gray, 1.3, 5)
    return faces
```

## model instance

```
#Create an instance from model detection
gender_model =cv2.dnn.readNetFromCaffe("gender.prototxt", "gender.caffemodel")
```

## setting classes labels

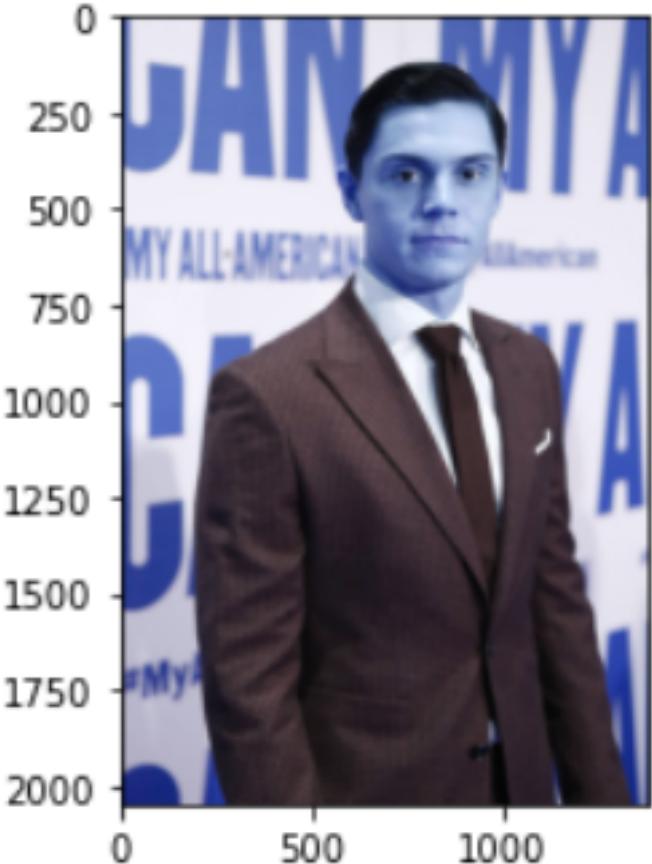
```
#Labels of the class
class_labels = ['Woman', 'Man']
```

# reading image



```
#Reading the first image
img_1 = cv2.imread('ep2.jpg')

#Showing the image
plt.imshow(img_1)
```



# Model employment



```
#Using the method implemented above
faces = detect_faces(img_1)
```

```
#Creating a loop to go through the image to detect the face and determine the gender
for face in faces:
    x, y, w, h = face

    #Cropping the detected face
    detected_face = img_1[int(y):int(y+h), int(x):int(x+w)]

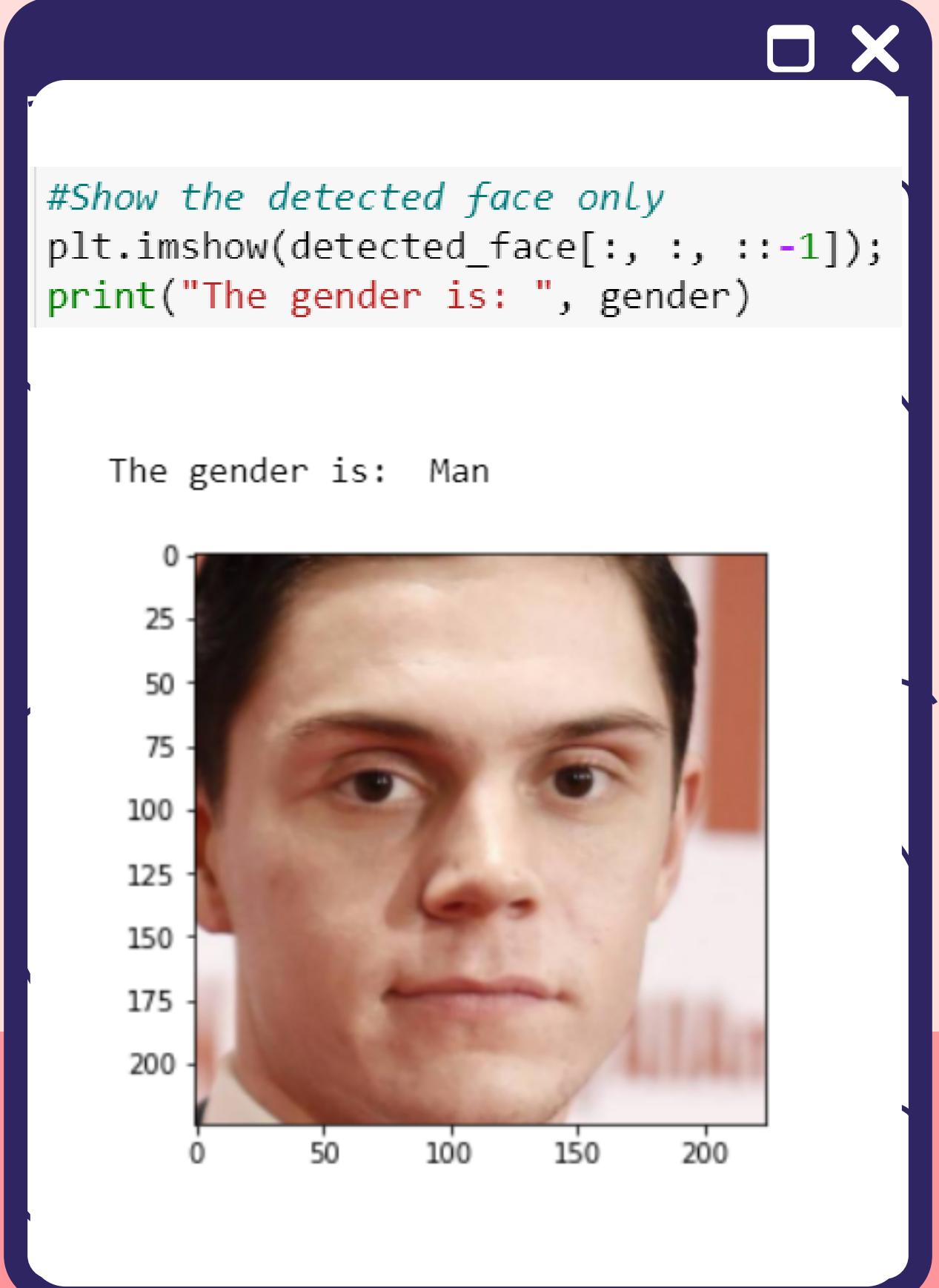
    #The caffe model that we are using expects (1, 3, 224, 224) input shape
    detected_face = cv2.resize(detected_face, (224, 224))
    #returns a blob which is our input image after mean subtraction, normalizing, and channel swapping
    imgG = cv2.dnn.blobFromImage(detected_face)

    #Using the model
    gender_model.setInput(imgG)

    #These are the classes we have in this model : Woman, Man
    #Runing a forward pass to compute the net output
    gender_class = gender_model.forward()[0]
    gender = 'Woman' if np.argmax(gender_class) == 0 else 'Man'
```

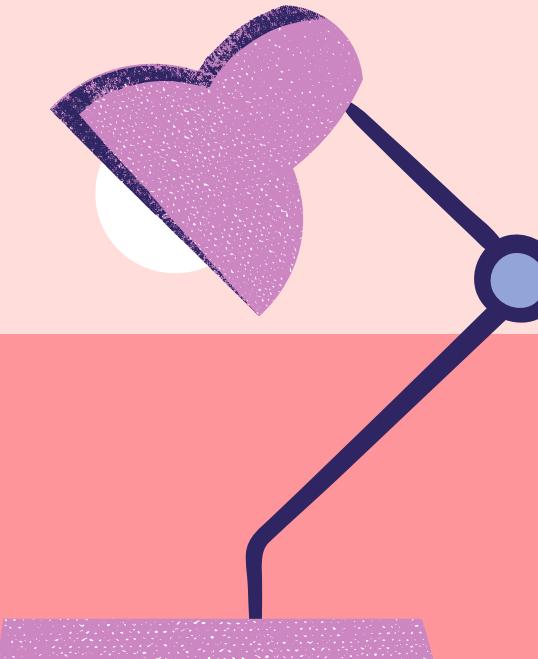
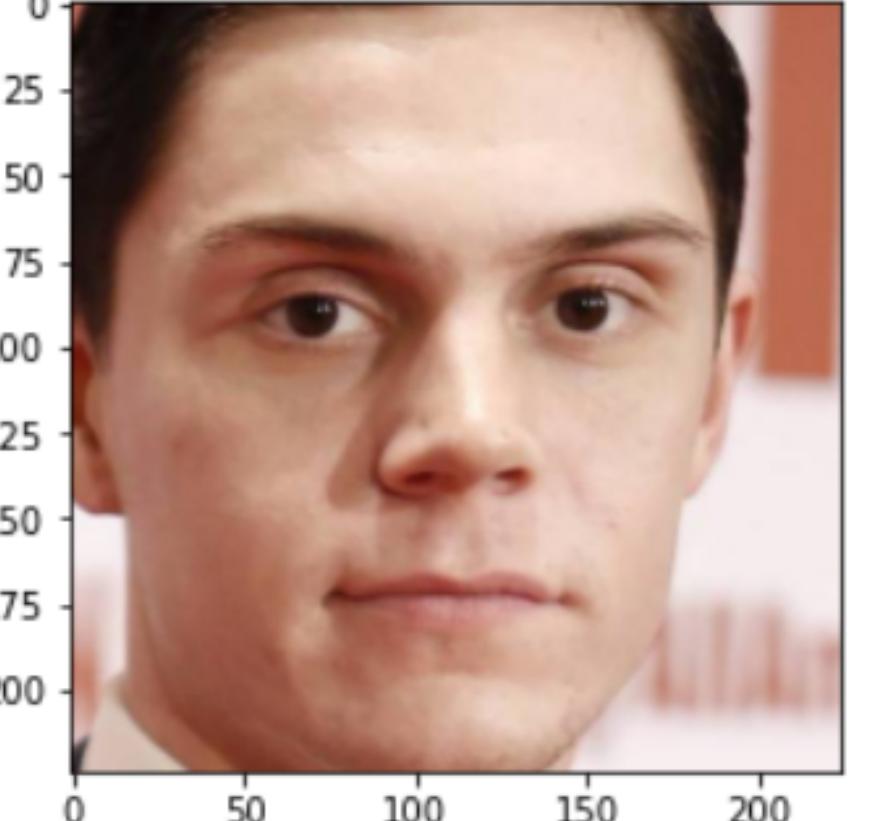


# detecting gender (Man)



```
#Show the detected face only
plt.imshow(detected_face[:, :, ::-1]);
print("The gender is: ", gender)
```

The gender is: Man



# detecting gender (Woman)



*#Showing the image*

```
plt.imshow(img_2)
```

*#Reading the second image*

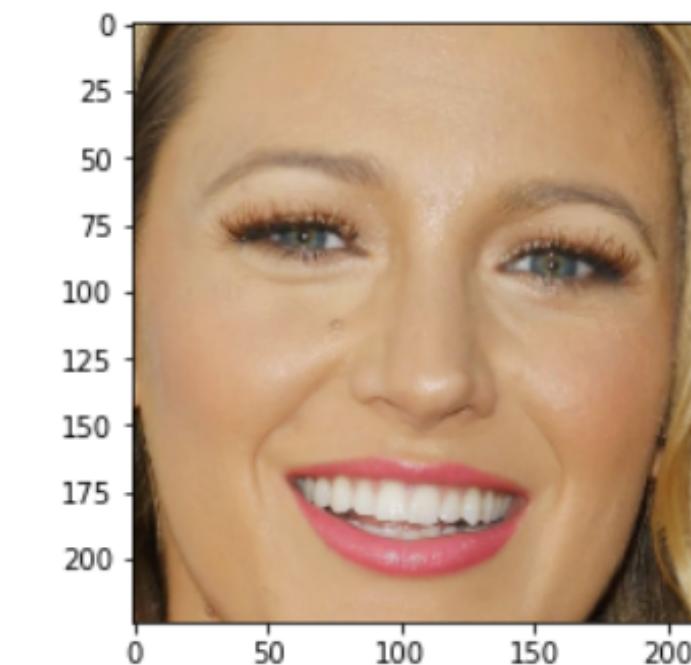
```
img_2 = cv2.imread('blake.jpg')
```



*#Show the detected face only*

```
plt.imshow(detected_face[:, :, ::-1]);  
print("Gender: ", gender)
```

Gender: Woman



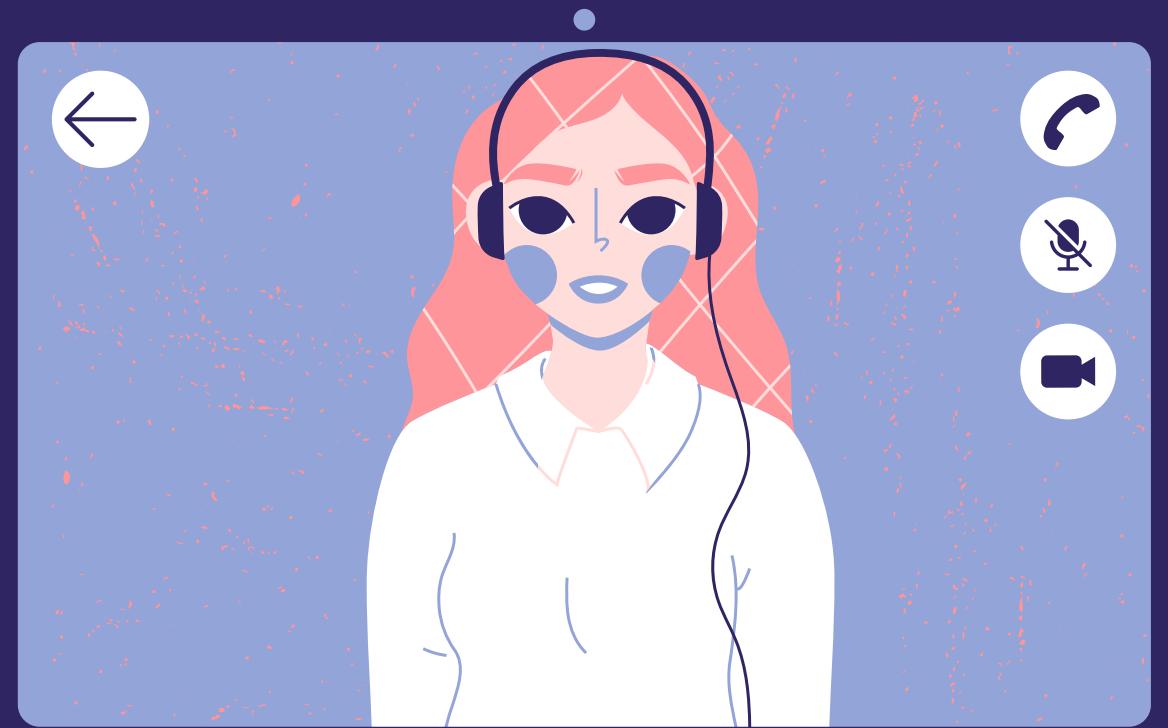
# Real-world Gender Detection Applications :

- A. video surveillance
- B. human-computer interaction
- C. customized advertisement



# Gender Detection Application for Company :

- A. · Understand what products and services different customer groups want and can afford.
- B. Target marketing campaigns more precisely and thus reduce the cost per lead or sale.
- C. Identify how society is changing and how they need to adapt.



# Thank you for listening





## References

- **Real-Time Gender Detection in the Wild Using Deep Neural Networks | IEEE Conference Publication | IEEE Xplore .**
- **Face Detection: Real-time applications with deep learning , at: <https://viso.ai/deep-learning/face-detection-overview/>**
- **Gender Detection using OpenCV in Python , at <https://www.thepythoncode.com/article/gender-detection-using-opencv-in-python>**
- **Age & Gender Detection: Top Use Cases <https://www.cameralyze.co/blog/age-gender-detection-top-use-cases>**