Here, the **blur\_faces()** function takes an image as input, detects faces using the pre-trained face detection model, and applies a Gaussian blur filter to the faces. The **blur\_faces()** function returns the original image with the faces blurred.

The **cv2.imread()** function is used to load the input image file. The **blur\_faces()** function is then called to process the image and generate the blurred image. Finally, the **cv2.imshow()** function is used to display the original and blurred images side by side. The **cv2.waitKey()** and **cv2.destroyAllWindows()** functions are used to wait for a key press and close the image windows when done.

Note that you will need to download the pre-trained face detection model and place it in the same directory as your Python script. You will also need to replace **example.jpg** with the path to your own input image file.

We can improve it with stream processing so we can blur any face that we don’t want to show such as if there is any women in life stream I can blur her face if she didn’t want to show for example .