Avinash Kumar - 1102244010

Assgn_04 » Techniques to prevent face identification

Here are two ways:

1. Blurring the ROI with kernel.
2. Mosaicing the ROI with zoom in and out

So, you try both methods as Assgn_04_1 and Assgn_04_2

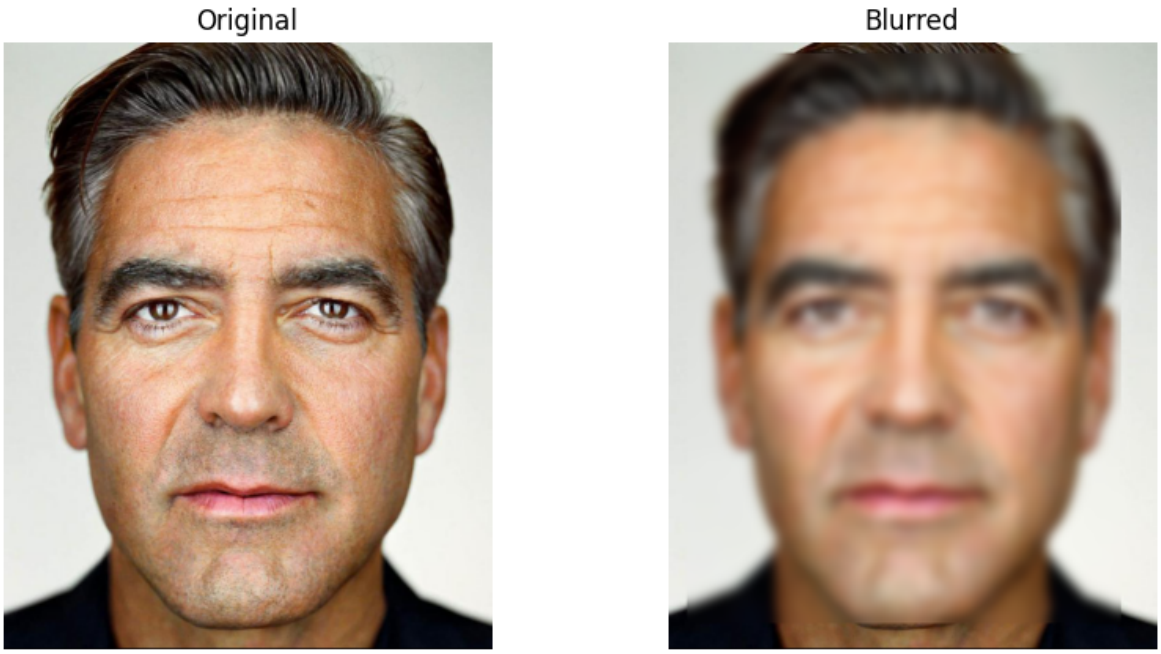
1. Assgn_04_1 Blurring on the ROI with kernel size

Drag with the mouse to blur the selected part with a blurring effect.

- After dragging the mouse, press Enter.

Please refer to the codes and output images below to fill in the blanks(codes)

In [1]:
```
import cv2
from matplotlib import pyplot as plt

ksize = 51  # The kernel size to use for blur handling.
win_title = 'blur'  # window title

img = cv2.imread('./images/practice_img/actor.jpg')  # read img
img_org = img.copy()  # make a copy for later use

while True:
    x, y, w, h = cv2.selectROI(win_title, img, fromCenter=False, showCrosshair
    if w > 0 and h > 0:  # Positive ROI
        roi = img[y:y + h, x:x + w]  # Designate ROI
        roi_blurred = cv2.GaussianBlur(roi, (ksize, ksize), 0)  # Blurring ROI
        img[y:y + h, x:x + w] = roi_blurred  # Apply blurred ROI to the origin
        cv2.imshow(win_title, img)
        key = cv2.waitKey(1) & 0xFF
        if key == 13:  # Enter key
            break
    else:
        break

cv2.destroyAllWindows()

fig, axs = plt.subplots(1, 2, figsize=(10, 5))
axs[0].imshow(cv2.cvtColor(img_org, cv2.COLOR_BGR2RGB))
axs[0].axis('off')
axs[0].set_title('Original')
axs[1].imshow(cv2.cvtColor(img, cv2.COLOR_BGR2RGB))
axs[1].axis('off')
axs[1].set_title('Blurred')

plt.show()
```



2. Assgn_04_2 Mosaicing the ROI with zoom in and

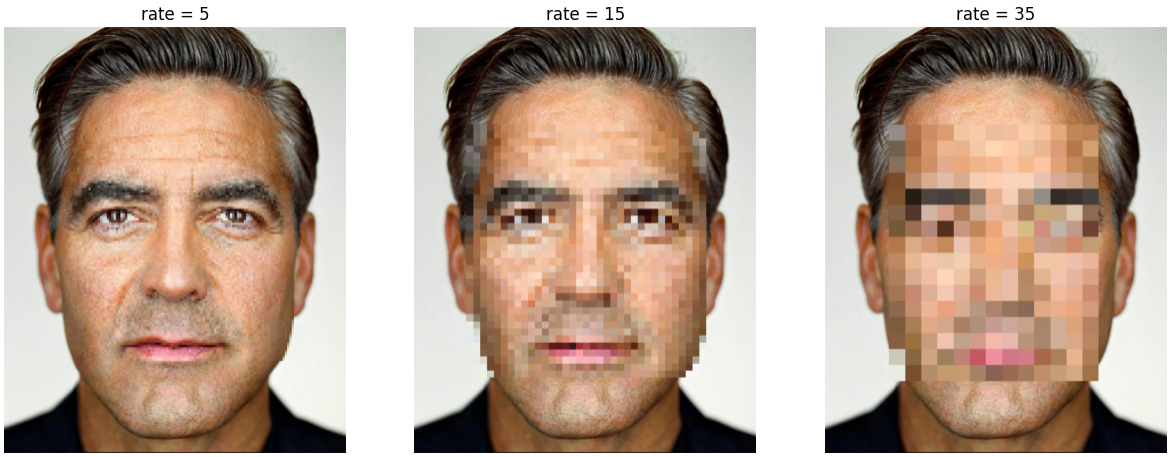
out

After shrinking(zoom out) the image of the ROI to which the mosaic is applied by a certain percentage,
and then zoom back in on the ROI.

- After selecting the ROI with dragging the mouse, press Enter.

In [9]:
```
import cv2
from matplotlib import pyplot as plt

rate = 35  # Shrink ratio to use for mosaic (1/rate)
win_title = 'mosaic'  # window title

img = cv2.imread('./images/practice_img/actor.jpg')  # read img

while True:
    x, y, w, h = cv2.selectROI(win_title, img, fromCenter=False, showCrosshair
    if w and h:
        roi = img[y:y + h, x:x + w]  # Designate ROI
        roi_small = cv2.resize(roi, (w // rate, h // rate), interpolation=cv2.
        roi_enlarged = cv2.resize(roi_small, (w, h), interpolation=cv2.INTER_N
        img[y:y + h, x:x + w] = roi_enlarged  # Apply to the original image
#         img5=img.copy()
#         img15=img.copy()
        img35=img.copy()
        cv2.imshow(win_title, img_copy)

    else:
        break

cv2.destroyAllWindows()

plt.imshow(cv2.cvtColor(img, cv2.COLOR_BGR2RGB))
plt.axis('off')
plt.title('Mosaic Applied Image')
plt.show()
```

In [10]:
```
fig, axs = plt.subplots(1,3, figsize=(15,10))
axs[0].imshow(cv2.cvtColor(img5, cv2.COLOR_BGR2RGB)), axs[0].axis('off'), axs[
axs[1].imshow(cv2.cvtColor(img15, cv2.COLOR_BGR2RGB)), axs[1].axis('off'), axs
axs[2].imshow(cv2.cvtColor(img35, cv2.COLOR_BGR2RGB)), axs[2].axis('off'), axs

plt.show()
```



In [ ]:
```

```