

Examen

Planteamiento del problema

Alterar una imagen proporcionada por el profesor, de tal forma que se obtenga de la forma más nítida posible una imagen del rostro. Se espera que cara de la imagen este de forma vertical mirando al frente.

Descripción de la solución

La resolución del examen se realizó en 4 pasos, el primero consiste en hacer un perfilado de la imagen, esto con el fin de aumentar la nitidez de la imagen y que se realice un perfilado. El segundo paso consiste en recortar la imagen del rostro, posteriormente en el tercer paso se aplica un filtro de sobel. Y por último rotamos la imagen para que quede de manera vertical.

Perfilado

El perfilado de la imagen lo aplique con un filtro llamado detailEnhance() que aparece en la documentación de Open CV, su función es resaltar los detalles de una imagen y después de varias pruebas con otros filtros como el GaussianBlur, Dilation, Median Blur, etc decidí utilizar este porque me dio los mejores resultados con la imagen.

```
cv.detailEnhance( src[, dst[, sigma_s[, sigma_r]]] ) -> dst
```

La función detailEnhance () recibe 4 parámetros los cuales son:

- src: la imagen a la que le aplicaremos el filtro.
- dst: la matriz donde guardaremos la imagen con el filtro ya aplicado.

Yo utilicé esta función en Python por lo que solo tuve que poner el src.

```
#Filtro para el perfilado de la imagen  
img = cv2.detailEnhance(src)
```

Recortar la imagen

Una vez aplicado el filtro anterior, decidí recortar la imagen del rostro, hice algunas pruebas por si había alguna diferencia entre recortar primero y aplicar el filtro después para ver si había un cambio de calidad notable, lo cual no hubo por lo que el orden lo dejé como esta.

Para recortar la imagen utilice un método de Numpy para recortar arreglos, que simplemente es definir la dimensión del corte, como la imagen tiene alto y ancho, la primera dimensión a recortar es la altura y la segunda es el ancho.

```
#Recortamos la imagen del rostro  
cropped_image = img[60:450, 100:550]
```

Y simplemente definí el corte con el que me sentí mas conforme.

Filtro de sobel

Ya había encontrado antes el filtro de sobel en la documentación de opencv por lo que no me fue difícil encontrar otra vez un ejemplo donde se aplicaba este filtro, por lo que tome el ejemplo y lo replique en el examen, adaptando por supuesto solo lo que necesitaba.

```
cv.Sobel( src, ddepth, dx, dy[, dst[, ksize[, scale[, delta[, borderType]]]] ) -> dst
```

La función de sobel recibe como parámetro lo siguiente:

- src: la imagen a la que le aplicaremos el filtro.
- ddepth: la profundidad de la imagen.
- dx: orden de la derivada de x.
- dy: orden de la derivada de y.
- dst: la matriz donde guardaremos la imagen con el filtro ya aplicado.
- ksize: tamaño del núcleo de sobel.
- scale: escala a la que queremos la imagen.
- delta: valor delta opcional que es añadido a la imagen resultante.
- borderType: tipo de borde en la imagen.

Rotar la imagen

En una tarea anterior se había pedido una rotación por lo que reutilice el código de esa tarea para usarlo aquí.

Lo primero que hice fue encontrar el centro de la imagen con la siguiente función:

```
#Obtenemos el centro de la imagen  
image_center = tuple(np.array(grad.shape[1::-1]) / 2)
```

Lo que se hace es que de la imagen hacemos un arreglo de la punta del inicio de la imagen al final y lo dividimos entre dos para obtener el centro.

```
#La rotamos -20 grados con respecto al centro  
rot_mat = cv2.getRotationMatrix2D(image_center, -20, 1.0)
```

Una vez hecho eso rotamos la imagen que obtenemos del centro -20° por lo que la rotación será hacia la izquierda, y conservamos la escala original.

```
#Combinamos la matriz grad con la que hemos rotado con la funcion warpAffine  
result = cv2.warpAffine(grad, rot_mat, grad.shape[1::-1], flags=cv2.INTER_LINEAR)
```

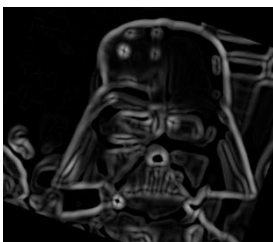
Para finalmente combinar la matriz original con la que esta rotada, mediante la función warpAffine () que aplica la transformación deseada en este caso la rotación en la imagen de destino.

Descripción de los resultados



El primer resultado que obtuve con el recorte y el filtro para ver más detalles en la imagen fue este como se ve la imagen se ve con mayor definición que la original y con el rostro notable a simple vista.

Posteriormente aplique el filtro de sobel para ver los resultados que arrojaba al aplicarlo primero que el recorte, después o al principio de todo, en algunas ocasiones modificando parámetros llegue a ver ligeros cambios menores en la calidad, pero nada significativo para que considerara cambiar el proceso.



La rotación al ser lo más simple decidí colocarla al final del proceso, ya que me parece que no hace diferencia alguna en la calidad de la imagen o en su defecto que se deprecie la imagen. Los únicos que para mi modificaban la imagen eran los primeros filtros y el recorte.

Discusión

Me parece interesante los cambios que puede sufrir una imagen al simplemente cambiar el orden en el que se aplican los filtros, cuando se podría pensar que esto no afectara en lo más mínimo, pero como vemos aquí si afecta mucho en la imagen.

Conclusión

En lo personal me fastidio un poco tener tan pocos cambios en la calidad de la imagen y tener que fijarse casi al milímetro en cual imagen daba un mejor resultado. Al principio decidí hacerlo tal cual venia en el examen, pero me daba una imagen terrible por lo que decidí experimentar paso por paso, primero hacer el recorte, luego el perfilado y ver que imagen resultaba, luego a esa la aplicaba el siguiente filtro y si en algún punto sentía que el resultado empeoraba regresaba y alteraba el orden, llego un punto en el que no note diferencia alguna y lo deje como me funciona. La documentación de openCV me sirvió de mucho apoyo para entender las funciones y como aplicarlas así como la comunidad que tiene en internet y varios foros.

Referencias

- <https://learnopencv.com/cropping-an-image-using-opencv/>
- https://docs.opencv.org/4.x/df/dac/group__photo__render.html#ga0de660cb6f371a464a74c7b651415975
- https://docs.opencv.org/4.x/d2/d2c/tutorial_sobel_derivatives.html
- <https://theailearner.com/tag/cv2-warpaffine/>