Reconocimiento de escenas
Proyecto sobre procesamiento de imágenes Prof.
Carlos B. Ogando M.

Indique a través de qué funciones y de qué librerías (ej. Opencv) se puede lograr realizar este proyecto sin implementar los algoritmos crudos.

Funciones útiles para el proyecto de la librería Opencv:

- **Gaussian Blur**

Applying Gaussian Blurring to an Image in OpenCV

We will now apply a Gaussian blur to an image, using OpenCV. This technique uses a Gaussian filter, which performs a weighted average, as opposed to the uniform average described in the first example. In this case, the Gaussian blur weights pixel values, based on their distance from the center of the kernel. Pixels further from the center have less influence on the weighted average. The following code convolves an image, using the **`GaussianBlur()`** function in OpenCV.

```
GaussianBlur(src, ksize, sigmaX[, dst[, sigmaY[, borderType]]])
```

```
"""
Apply Gaussian blur
"""
# sigmaX is Gaussian Kernel standard deviation
# ksize is kernel size
gaussian_blur = cv2.GaussianBlur(src=image, ksize=(5,5), \\
sigmaX=0, sigmaY=0)

cv2.imshow('Original', image)
cv2.imshow('Gaussian Blurred', gaussian_blur)

cv2.waitKey()
cv2.imwrite('gaussian_blur.jpg', gaussian_blur)
cv2.destroyAllWindows()
```

- **Sobel Filter**

The following is the syntax for applying Sobel edge detection using OpenCV:

```
Sobel(src, ddepth, dx, dy)
```

```
# Sobel Edge Detection
sobelx = cv2.Sobel(src=img_blur, ddepth=cv2.
Detection on the X axis
sobely = cv2.Sobel(src=img_blur, ddepth=cv2.CV_64F, dx=0, dy=1, ksize=5) # Sobel Edge
Detection on the Y axis
sobelxy = cv2.Sobel(src=img_blur, ddepth=cv2.CV_64F, dx=1, dy=1, ksize=5) # Combined X
and Y Sobel Edge Detection
```

- **Edge Detection**

The following is the syntax for applying Canny edge detection using OpenCV:

```
Canny(image, threshold1, threshold2)
```

```
# Canny Edge Detection
edges = cv2.Canny(image=img_blur, threshold1
# Display Canny Edge Detection Image
cv2.imshow('Canny Edge Detection', edges)
cv2.waitKey(0)
```

Fuente: https://learnopencv.com/image-filtering-using-convolution-in-opencv/#gauss-blur-opencv

Indique las aplicaciones en la vida real de este proyecto. Favor ilustrar con un caso de estudio.

Edge Detection

Algunas de las aplicaciones más comunes de la **extracción de bordes** son:

- **Reconocimiento de huellas dactilares:** al reconocer huellas dactilares, es útil preprocesar la imagen mediante la detección de bordes. En este caso, los "bordes" son los contornos de la huella dactilar, en contraste con el fondo sobre el que se hizo la huella dactilar. Esto ayuda a reducir el ruido para que el sistema pueda enfocarse exclusivamente en la forma de la huella digital.

- **Imágenes médicas:** al igual que el reconocimiento de huellas dactilares, las imágenes médicas son otro campo en el que los sistemas de visión por computadora pueden mejorar el rendimiento al eliminar el ruido y la información extraña y mejorar la atención médica con IA. Las imágenes médicas son susceptibles a diferentes tipos de ruido durante el proceso de recopilación de datos. La aplicación de la detección de bordes facilita que los sistemas de visión por computadora (y los médicos humanos) detecten anomalías en las imágenes médicas.

- **Detección de vehículos:** Considere uno de los casos de uso de visión artificial más avanzados: los automóviles autónomos. Estos sistemas dependen de la capacidad de detectar rápida y automáticamente otros vehículos en la carretera. Esta tecnología puede ayudar a reducir drásticamente la complejidad de las imágenes que necesitan analizar los vehículos autónomos, al mismo tiempo que conserva las formas reconocibles de otros vehículos.

Hybrid Images:

La fusión de imágenes juega un papel muy importante en el diagnóstico médico para ayudar a los médicos a examinar las anomalías en las imágenes de TC y RM. En este capítulo, se analizaron diferentes técnicas de fusión de imágenes y se implementaron tres algoritmos en MATLAB para dos conjuntos de datos diferentes recopilados de varias fuentes y se compararon los resultados con los métodos existentes en la literatura. Las imágenes de tomografía computarizada de baja dosis se han mejorado mediante técnicas de mejora de imagen y se han fusionado con imágenes de resonancia magnética. A partir de los resultados se analiza que el algoritmo PCA resulta en un mejor desempeño en cuanto a relación pico de señal a ruido, error cuadrático medio de la raíz y error cuadrático medio de la raíz

Fuentes: https://chooch.ai/computer-vision/what-are-the-applications-of-edge-detection/#:~:text=Applying%20edge%20detection%20makes%20it,cases%3A%20self%2Ddriving%20cars.

https://www.intechopen.com/chapters/76249