



Reporte Loading Video Source

Nombre: Edgar Javier Fregoso Cuarenta

Registro: 22310285

Materia : visión Artificial

Importación de bibliotecas: Se importan las bibliotecas necesarias, como numpy y cv2 de OpenCV.^[?]

1. `import numpy as np`
2. `import cv2`

^[?]

2. **Captura de video:** Se crea un objeto VideoCapture para acceder a la cámara web o a un archivo de video. El argumento 0 indica que se utilizará la primera cámara conectada al sistema.^[?]

3. `cap = cv2.VideoCapture(0)`

^[?]

3. **Lectura y procesamiento de fotogramas:** Se inicia un bucle que lee cada fotograma del video. La función `cap.read()` devuelve dos valores: `ret`, un booleano que indica si la lectura fue exitosa, y `frame`, que contiene el fotograma.^[?]

4. `while(True):`
5. `ret, frame = cap.read()`
6. `gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)`
7. `cv2.imshow('frame', gray)`
8. `if cv2.waitKey(1) & 0xFF == ord('q'):`
9. `break`

^[?]

En este ejemplo, cada fotograma se convierte a escala de grises utilizando `cv2.cvtColor` y se muestra en una ventana llamada 'frame'. El bucle continúa hasta que se presiona la tecla 'q'.

4. **Liberación de recursos:** Después de finalizar el bucle, se liberan los recursos asociados al video y se cierran las ventanas abiertas.^[?]

5. `cap.release()`
6. `cv2.destroyAllWindows()`

^[?]

5. **Grabación de video:** El tutorial también aborda cómo guardar el video procesado en un archivo. Se utiliza `cv2.VideoWriter` para definir el códec, el nombre del archivo de salida, la tasa de fotogramas y el tamaño de los fotogramas.^[2]

6. `fourcc = cv2.VideoWriter_fourcc(*'XVID')`

7. `out = cv2.VideoWriter('output.avi', fourcc, 20.0, (640, 480))`

^[2]

Dentro del bucle, se escribe cada fotograma en el archivo de salida:

```
out.write(frame)
```

^[2]

Al finalizar, se liberan tanto el objeto de captura como el de escritura:

```
cap.release()
```

```
out.release()
```

```
cv2.destroyAllWindows()
```

^[2]