



Rodrigo Hernández Zavala

15 de marzo de 2025

Ing. Mecatrónica

22310215

Efectos en Video

Visión Artificial

Explicación del Funcionamiento de la Práctica (Efectos en Video con OpenCV)

Esta práctica captura video en **tiempo real** desde la cámara del sistema y aplica diferentes **efectos de procesamiento de imagen** utilizando **OpenCV**.

Paso a Paso del Código

```
python
CopiarEditar
import cv2
import numpy as np
```

◆ **Importa las librerías necesarias:**

- `cv2` → Para capturar video y procesar imágenes.
 - `numpy` → Para manipular matrices de píxeles.
-

```
python
CopiarEditar
cap = cv2.VideoCapture(0)
```

◆ **Inicia la captura de video desde la cámara predeterminada (0).**

- Si hay otra cámara externa, se puede probar con 1, 2, etc.
-

```
python
CopiarEditar
while True:
    ret, frame = cap.read()
    if not ret:
        break
```

◆ **Captura cada fotograma del video en un bucle infinito.**

- ◆ `ret` verifica si la captura fue exitosa (`True` = cámara funcionando).
-

```
python
CopiarEditar
gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
blur = cv2.GaussianBlur(frame, (15, 15), 0)
edges = cv2.Canny(frame, 100, 200)
inverted = cv2.bitwise_not(frame)
mirror = cv2.flip(frame, 1)
```

◆ **Aplica diferentes efectos:**

1. **Escala de Grises** → `cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)`
2. **Desenfoque** → `cv2.GaussianBlur(frame, (15, 15), 0)`
3. **Detección de Bordes** → `cv2.Canny(frame, 100, 200)`
4. **Negativo (Inversión de Colores)** → `cv2.bitwise_not(frame)`
5. **Espejo (Reflejo Horizontal)** → `cv2.flip(frame, 1)`

```
python
CopiarEditar
cv2.imshow('Original', frame)
cv2.imshow('Escala de Grises', gray)
cv2.imshow('Desenfoque', blur)
cv2.imshow('Bordes', edges)
cv2.imshow('Negativo', inverted)
cv2.imshow('Espejo', mirror)
```

◆ **Muestra cada efecto en una ventana separada.**

- ◆ Se pueden cerrar con la tecla 'q'.

```
python
CopiarEditar
if cv2.waitKey(1) & 0xFF == ord('q'):
    break
```

◆ **Espera a que el usuario presione 'q' para salir del bucle y cerrar las ventanas.**

```
python
CopiarEditar
cap.release()
cv2.destroyAllWindows()
```

◆ **Libera la cámara y cierra todas las ventanas de OpenCV.**

Objetivos de la Práctica

- ✓ **Capturar video en tiempo real desde la cámara del sistema.**
 - ✓ **Aplicar efectos básicos de procesamiento de imágenes con OpenCV.**
 - ✓ **Comprender cómo transformar imágenes usando filtros y detección de bordes.**
 - ✓ **Practicar la manipulación de matrices de píxeles para crear efectos visuales.**
-



Posibles Mejoras

- ◆ **Permitir cambiar de efecto con el teclado**
 - Podemos usar `cv2.waitKey()` para alternar entre diferentes filtros con teclas específicas.
 - ◆ **Guardar el video con los efectos aplicados**
 - Usando `cv2.VideoWriter()` para exportar el video procesado.
 - ◆ **Aplicar detección de rostros en tiempo real**
 - Combinando `cv2.CascadeClassifier()` con los efectos.
 - ◆ **Agregar más filtros avanzados**
 - Como efecto térmico, modo noche, o filtros personalizados con `cv2.LUT()`.
-



Conclusión

- ✦ **La práctica demuestra cómo aplicar filtros en tiempo real a un video capturado con OpenCV.**
- ✦ **Ayuda a comprender cómo manipular imágenes en forma de matrices.**
- ✦ **Es una base excelente para proyectos avanzados como realidad aumentada o visión artificial.**