



## ***Visión Artificial***

***No. de Practica: 2***

***Título: Loading Video Source OpenCV Python  
Tutorial***

***Nombre: Aldo Misael Osuna Rodríguez***

***Registro: 22310221  
6°G***

***18-marzo-2025***

## Objetivo:

En este tutorial de OpenCV con Python, cubriremos algunas operaciones básicas con vídeo y cámaras web.

## Código:

```
import numpy as np
import cv2

cap = cv2.VideoCapture(0)
fourcc = cv2.VideoWriter_fourcc(*'XVID')
out = cv2.VideoWriter('output.avi',fourcc, 20.0, (640,480))

while(True):
    ret, frame = cap.read()
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    out.write(frame)
    cv2.imshow('frame',gray)
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break

cap.release()
out.release()
cv2.destroyAllWindows()
```

## Comentarios:

Primero importamos "numpy" and "cv2" y despues en nuestro codigo de python cambiamos en cv2.VideoCapture(1), de 1 a 0 dentro del parentesis de VideoCapture(0). Esto es porque al tener dentro de VideoCapture() un numero 1 estamos mandando a llamar a una camara secundaria/externa, entonces al cambiar este 1 por un 0 estamos activando a nuestra camara principa que es la que esta integrada en nuestro dispositivo.

Este código inicia un bucle infinito (que se romperá posteriormente con una sentencia break), donde ret y frame se definen como cap.read(). Básicamente, ret es un booleano que indica si hubo un retorno, y frame es cada fotograma devuelto. Si no hay ningún fotograma, no se generará un error, sino None.

while(True):

    ret, frame = cap.read()

Aquí, definimos una nueva variable, gris, como el marco, convertido a gris. Observe que indica BGR2GRAY. Es importante tener en cuenta que OpenCV interpreta los colores como BGR (Azul, Verde, Rojo), mientras que la mayoría de las aplicaciones informáticas los interpretan como RGB (Rojo, Verde, Azul).

gray = cv2.cvtColor(frame, cv2.COLOR\_BGR2GRAY)

Tenga en cuenta que, a pesar de ser una transmisión de video, seguimos usando imshow. Aquí, mostramos la señal convertida a gris. Si desea mostrar ambas simultáneamente, puede usar imshow para el fotograma original e imshow para el gris, y así aparecerán dos ventanas.

```
cv2.imshow('frame',gray)
```

Esta sentencia solo se ejecuta una vez por fotograma. Básicamente, si obtenemos una clave, y esa clave es una q, saldremos del bucle while con un break, que luego se ejecuta:

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

Esto libera la cámara web y cierra todas las ventanas de imshow().

En algunos casos, es posible que quieras grabar y guardar la grabación en un nuevo archivo.

```
cap.release()
```

```
cv2.destroyAllWindows()
```

## Resultados:

