

Laboratorio 2

1. Competencias a desarrollar

En este laboratorio el estudiante reforzará el uso de procesamiento de video, kernels y optimización con Python y OpenCV. El estudiante entregará un script por cada tarea en la actividad correspondiente en **MiU** antes del **miércoles 18 de octubre a las 23:59**.

2. Instrucciones

Las tres tareas se centrarán en la aplicación de filtros (inspirados en Instagram) utilizando OpenCV y aplicados a videos. Como referencia se utilizará este [sitio](#) que muestra algunos tipos básicos de filtros y cómo implementarlos.

a Tarea 1: HDR o Pencil

Programe un script que implemente el filtro HDR effect o Pencil Sketch Effect: Colour mostrados en el artículo de referencia (Puede utilizar las funciones nativas de OpenCV mencionadas en la referencia.).

Su script debe contener su nombre comentado en la primera línea, ejecutarse como `python tarea_1.py` y debe aplicar el efecto seleccionado a la imagen que proviene de una webcam detectada y mostrar el output en una ventana.

b Tarea 2: Sharp Effect

Programe un script que implemente el filtro Sharp Effect mostrado en el artículo de referencia utilizando su propio código para aplicar el kernel (**No puede** utilizar las funciones nativas de OpenCV mencionadas en la referencia).

Su script debe contener su nombre comentado en la primera línea, ejecutarse como `python tarea_2.py` y debe aplicar el efecto a la imagen que proviene de una webcam detectada y mostrar el output en una ventana.



Efecto Sharp

c Tarea 3: Fast Sharp Effect o Personalizado

Mejore el desempeño de su filtro Sharp Effect que programó en la tarea 2 utilizando Cython. Opcionalmente puede implementar su propio efecto e implementar una versión con mejor desempeño utilizando Cython.

Agregue un comentario dentro del docstring de su función, explicando el speedup obtenido Vs. La implementación de la tarea 2 utilizando un video .mp4 como material de prueba.

$$\text{Speedup} = \frac{t_{\text{sharp}}}{t_{\text{fast-sharp}}}$$

Su script debe contener su nombre comentado en la primera línea, ejecutarse como `python tarea_3.py` y debe aplicar el efecto a la imagen que proviene de una webcam detectada y mostrar el output en una ventana.

3. Punteo

- Tarea 1: 33%
- Tarea 2: 33%
- Tarea 3: 33%