



UNIVERSIDAD AUSTRAL DE CHILE INSTITUTO DE ELECTRICIDAD Y ELECTRÓNICA

TAREA II

VISIÓN ARTIFICIAL Y REDES NEURONALES (ELEP 233)

**Todos los programas deben incluir comentarios.
Incluya un archivo .pdf con las comparaciones solicitadas en la tarea.**

1. Escriba un programa en Python que sea capaz de recortar una región de una imagen de la manera descrita en clases. El programa debe solicitar como dato de entrada los números de fila y columna correspondientes a los vértices superior izquierdo e inferior derecho de la región a recortar. Además, el programa debe autocorregir la posición de la región de recorte en caso de que el usuario la posicione fuera de los márgenes de la imagen o no ingrese los vértices en el orden solicitado. Almacene la imagen a recortar en la variable *img*.

2. Cree un programa en Python que sea capaz de realizar un acercamiento digital de una imagen RGB, almacenada en la variable *img*, mediante el método presentado en clases. El programa debe solicitar como dato de entrada el factor del zoom (un número entero entre 1 y 10. Por ejemplo, 2 significa que el zoom se realiza 2 veces). Compare con la función *resize* de OpenCV.

3. Realice operaciones algebraicas en imágenes RGB. **Cuide que los valores resultantes de cada píxel no superen el valor máximo ni sean inferiores al valor mínimo permitido en la matriz de imagen.**

a) Escoja dos imágenes RGB distintas de la misma dimensión y escriba un código en Python que las sume. Compare con las funciones *add* y *addWeighted* de OpenCV.

b) Escriba un código en Python que reste las 2 imágenes anteriores. Compare con la función *subtract* de OpenCV.

c) Escriba un código en Python que multiplique una de las imágenes por una constante (mayor que 1), aproximando resultados decimales al entero más próximo. Comente acerca del efecto de esta operación y compare con la función *multiply* de OpenCV.

d) Escriba un código en Python que divida una de las imágenes por una constante (mayor que 1), aproximando resultados decimales al entero más próximo. Comente acerca del efecto de esta operación y compare con la función *divide* de OpenCV.

4. Programe el filtro promedio y el filtro medio a aplíquelos a una imagen contaminada con ruido “sal y pimienta” (utilice la función *noisy* de Python para generar el ruido). Compare el resultado obtenido con la función *medianBlur* de OpenCV.
5. Programe en Python la especificación de histograma. Pruebe su programa con 3 diferentes tipos de histograma y comente cómo afecta a la imagen el histograma especificado.