

Taller de Visión por Computador

Curso de Computer Vision – Pregrado

Objetivo

El propósito de este taller es que los estudiantes desarrollen una librería en Python llamada `cvtools`, con módulos organizados para tareas fundamentales de visión por computador. Cada módulo debe contener funciones reutilizables que puedan importarse y probarse desde un script principal.

Estructura esperada del repositorio

`computer-vision-taller/`

<code>cvtools/</code>	<code># Librería principal</code>
<code> __init__.py</code>	
<code> camera.py</code>	<code># Modelo de cámara y proyecciones</code>
<code> color.py</code>	<code># Espacios de color y cuantización</code>
<code> filters.py</code>	<code># Convoluciones y filtros clásicos</code>
<code>tests/</code>	<code># Tests unitarios por módulo</code>
<code> test_camera.py</code>	
<code> test_color.py</code>	
<code> test_filters.py</code>	
<code>data/</code>	<code># Carpeta con imágenes de prueba</code>
<code> ejemplo1.jpg</code>	
<code> ejemplo2.png</code>	
<code>main.py</code>	<code># Script demostrativo que importa funciones</code>
<code>README.md</code>	<code># Explicación del taller y cómo usar la librería</code>
<code>requirements.txt</code>	<code># Librerías necesarias</code>

Ejercicios

Módulo `camera.py`

1. Implementar una función para aplicar **distorsión radial** a puntos proyectados, utilizando parámetros k_1, k_2 .
2. Implementar una función para variar la **longitud focal** y observar el cambio en la perspectiva de los puntos proyectados.

Módulo `color.py`

1. Implementar funciones para convertir imágenes entre los espacios de color: **RGB** \rightarrow **HSV** y **RGB** \rightarrow **LAB**.
2. Implementar una función que calcule y grafique el **histograma de colores** de una imagen.
3. Implementar una función de **cuantización simple** que reduzca la imagen a un número limitado de colores (ej. 256, 64, 16).
4. Implementar una función que **reduzca el peso de la imagen** disminuyendo la cantidad de colores y que retorne el nuevo tamaño en KB.

Módulo `filters.py`

1. Implementar una función de **convolución genérica** que reciba una imagen y un kernel.
2. Implementar funciones para aplicar el **filtro Sobel en X y en Y**.
3. Implementar una función que aplique el **detector de Canny**.
4. Implementar un **filtro Laplaciano** y documentar qué resalta en la imagen resultante.

Entrega

El estudiante deberá entregar:

- La carpeta `cvtools/` con los módulos implementados.
- Un script `main.py` que muestre ejemplos de uso de cada función.
- Tests unitarios simples en la carpeta `tests/`.
- Se debe entregar el link del repositorio por medio de la plataforma academica a más tardar el 29 de agosto 2025 hasta las 11:59pm.