

**18 Detección de rectángulos**

**Tutorial Procesamiento de Imagen con webcam**

**Estudiantes Practicantes:**

Luis Pereira

**Profesor:**

Luis Vera

**Laboratorio CIMUBB**

2023-2

**Importación de bibliotecas**

Primero, debes importar las bibliotecas necesarias. En este código, utilizamos cv2 (OpenCV) y numpy.



**cv2**: Es la biblioteca OpenCV que utilizamos para capturar video desde la cámara, procesar imágenes y dibujar contornos.

**numpy**: Se usa para realizar operaciones matemáticas en matrices y arreglos.

**Inicialización de la cámara y la ventana**

**Texto

Descripción generada automáticamente**

**cv2.VideoCapture(0)**: Inicializa la cámara predeterminada. Puedes cambiar el número entre paréntesis si tienes múltiples cámaras conectadas.

**cv2.namedWindow('Detención de Rectángulos', cv2.WINDOW\_NORMAL):** Crea una ventana llamada 'Detención de Rectángulos' para mostrar los resultados. El segundo argumento **cv2.WINDOW\_NORMAL** establece la ventana como redimensionable.

**Bucle Principal**

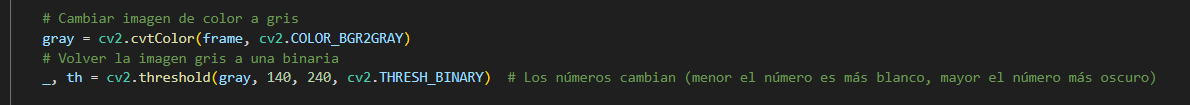
**Imagen que contiene Interfaz de usuario gráfica

Descripción generada automáticamente**

Un bucle **while** se ejecuta continuamente.

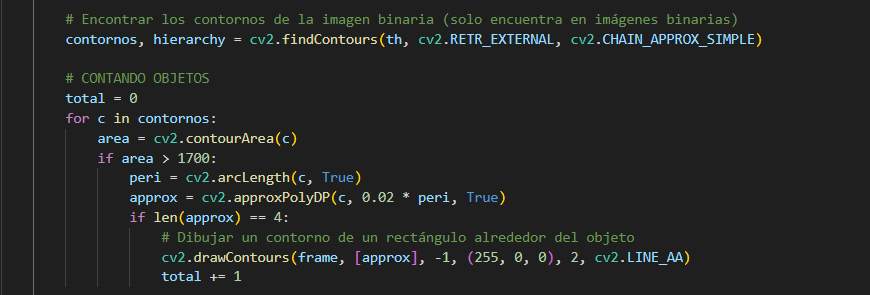
**cap.read()**: Lee un fotograma de la cámara. La variable frame contiene la imagen capturada.

**Procesamiento de Imagen**

****

Convertimos la imagen a escala de grises y luego a una imagen binaria. Esto facilita la detección de contornos.

**Detección de Contornos y Rectángulos**

****

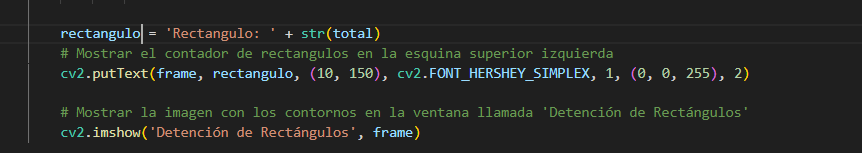
Usamos cv2.findContours para encontrar los contornos en la imagen binaria.

Filtramos los contornos que tienen un área mayor que 1700.

Identificamos los contornos que tienen 4 vértices (posibles rectángulos) utilizando cv2.approxPolyDP.

Dibujamos un contorno alrededor de cada rectángulo encontrado.

**Mostrar la imagen**

****

Agregamos un texto al fotograma para mostrar el número de rectángulos detectados.

Mostramos la imagen procesada en la ventana 'Detención de Rectángulos'.

**Detener la ejecución**

Texto

Descripción generada automáticamente

Esperamos a que el usuario presione la letra ‘q’ para finalizar el programa.

**Liberar recursos**

**Interfaz de usuario gráfica, Texto

Descripción generada automáticamente con confianza media**

Liberamos la cámara y cerramos todas las ventanas cuando el programa finaliza.

**Resultado Final**

**Forma

Descripción generada automáticamente con confianza baja**