**Práctica 9**

**Segmentación**

A screenshot of a computer code

Description automatically generated**Describe un algoritmo que determine el número de objetos de una imagen, como se muestra a continuación, prueba con otra imagen:**

A close-up of a blue and white background

Description automatically generated

El algoritmo por la consola de comandos devuelve un total de 12 objetos detectados.

**Realiza los siguientes apartados:**

1. A screenshot of a computer program

   Description automatically generated**Determina las componentes conectadas de ESTA IMAGEN y el centro de cada una de las regiones correspondientes.**

**A black and white image of circles

Description automatically generated**

1. **¿Por qué se detectan varios centros en una de las monedas?**

El umbral calculado por graythresh puede no ser el óptimo para todas las monedas, resultando en una segmentación incorrecta.

1. **¿Qué ocurre si cambias bwlabel(B,8) por bwlabel(B,4)? ¿se detectan las mismas componentes conectadas y objetos? ¿Por qué?**

Con conectividad 4, podrías detectar más componentes conectadas si los objetos en la imagen tienen conexiones delgadas o están conectados diagonalmente. Estos se verían como componentes separadas.

1. **Prueba a detectar el centro de las componentes conectadas de otra imagen y comenta el resultado.**

No hay muchas diferencias más que solo existe un único centro justo en el centro de la imagen.

A close up of a black and white image

Description automatically generated

A screenshot of a computer program

Description automatically generated

**Estás diseñando un sistema automático para que el vehículo no se salga del carril en los tramos rectos. Tu primera tarea es identificar las líneas de la carretera, describe un algoritmo que detecte las 3 líneas más largas de la IMAGEN PROPUESTA.**

**a) Prueba a aumentar el número de líneas a detectar y muestra el resultado.**

No se pueden detectar más líneas puesto que la imagen no ha sido proporcionada.

**b) Prueba a detectar líneas en otra imagen y comenta el resultado.**

Se utiliza la función houghpeaks para encontrar los picos en la matriz de acumulación de la transformada de Hough. Luego, estos picos se procesan para obtener las líneas correspondientes mediante la función houghlines. Se filtran las líneas detectadas conservando solo las más largas (si modificamos la variable de *numLineas* detectara más o menos líneas). Estas líneas se dibujan en la imagen original junto con sus puntos finales y el segmento más largo se destaca en rojo.

**A screenshot of a computer code

Description automatically generated**

**A black and white image of a city

Description automatically generated**