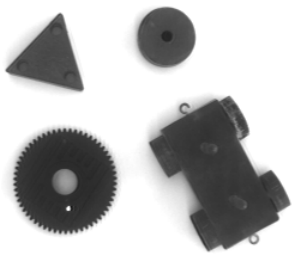
Trabajo 2. Reconocimiento 2D

El objetivo es desarrollar con OpenCV un programa capaz de reconocer objetos planos a partir de muestras previamente aprendidas.

Pasos a seguir:

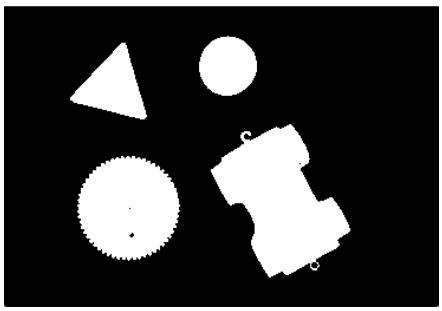
1. **Umbralización** de las imágenes. Las imágenes originales son imágenes en blanco y negro, como la siguiente:



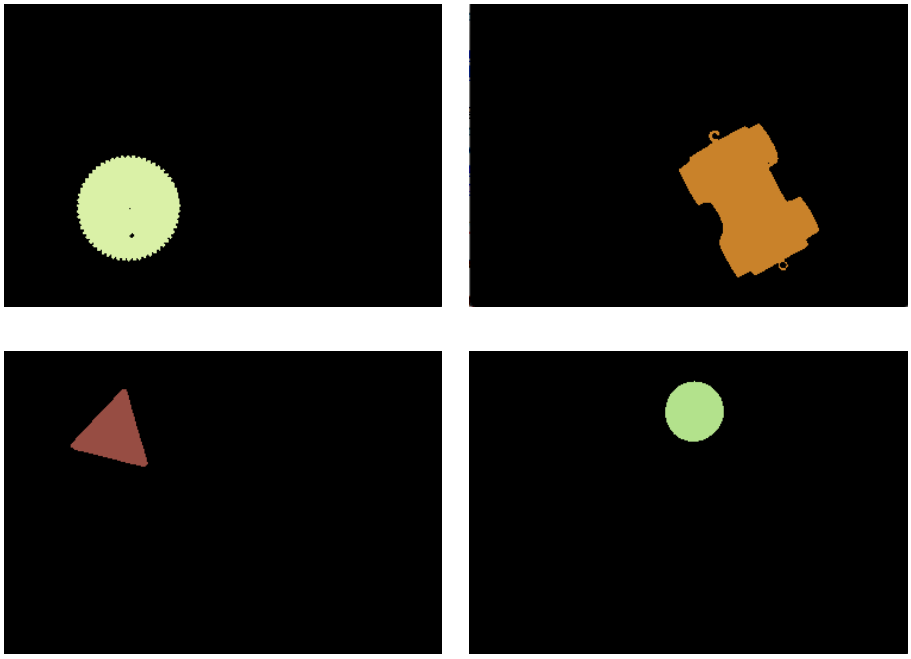
Debes convertirlas a imágenes binarias. Para ello, debes decidir la umbralización más adecuada entre el método de Otsu y el método adaptativo, ambos disponibles en OpenCV.

(http://docs.opencv.org/trunk/doc/py_tutorials/py_imgproc/py_thresholding/py_thresholding.html?highlight=otsu)

El resultado debe así (objetos blancos, fondo negro):



2. A partir de las imágenes binarias, debe obtener los blobs que corresponden a diferentes objetos que aparezcan en la imagen. Para ello puedes utilizar la función findContours. (http://docs.opencv.org/modules/imgproc/doc/structural_analysis_and_shape_descriptors.html?highlight=findcontours#cv2.findContours)



3. **Obtención de parámetros descriptores** de cada blob o objeto que aparezca en una imagen**.** A partir de la segmentación en contornos de una imagen, debes determinar para cada contorno su área en píxeles (a partir de sus momentos de imagen), su perímetro (a partir del contorno) y los tres primeros momentos invariantes (tienes funciones en openCV para todo, no vale la pena programar los cálculos específicos).

4. **Aprendizaje o Entrenamiento supervisado**: para cada uno de los objetos que el sistema debe reconocer (vagón, rectángulo, triángulo, círculo, rueda), debes aprender la **media y varianza** de cada parámetro o descriptor. Escribe el programa `aprender nomfich nomobj`, cuyos dos argumentos son:

   - `Nomfich:` nombre del fichero en el que esté almacenada la imagen de la muestra del objeto a aprender

   - `Nomobj:` nombre del objeto que aparece en la imagen.

El programa debe actualizar en el fichero `objetos` los datos de los objetos que conoce (sugerencia: utiliza FileStorage).

5. **Reconocimiento**: escribe el programa `reconocer nomfich`, cuyo argumento es el fichero que contiene la imagen de prueba (`reco1.pgm, reco2.pgm, reco3.pgm`). Dada la imagen en la que aparecen varios objetos, el sistema debe intentar el reconocimiento de cada uno por el método de las distancias mínimas de Mahalanobis. Si un objeto pasa el test de Mahalanobis con una sola clase, el sistema declarará que se trata de esa clase. Si pasa el test con más de un objeto, declarará que tiene dudas entre las clases. Si no pasa el test con ninguna clase, declarará que el objeto es desconocido.

6. **Regularización**: determina si el sistema tiene falsos positivos/negativos. En ese caso, compensa el pequeño número de muestras en el cálculo de la varianza y prueba utilizar diferentes valores para el porcentaje de la media.