**Atividade 6: Processamento de Imagem para Extração de Contornos**

O motivo desse relatório é descrever o funcionamento do processamento de imagem utilizando a biblioteca OpenCV para extrair os contornos externos e internos de determinadas elementos de imagens fornecidas pelo professor.

O objetivo do processamento é separar as silhuetas de elementos de uma determinada imagem a partir da aplicação de filtros, conversão de imagem e operações morfológicas. Este relatório descreve detalhadamente os passos executados, sendo eles respectivamente:

1. **Carregamento e Conversão da Imagem:**  
   A imagem é carregada em formato JPEG que depois é convertida para BGR (para o OpenCV interpretar) e convertida para RGB para facilitar a visualização com bibliotecas de plotagem, como o Matplotlib.
2. **Conversão para Escala de Cinza (Gray Scale):**  
   A conversão para escala de cinza é realizada para reduzir a complexidade da imagem, permitindo o processamento com técnicas de threshold (binarização).
3. **Aplicação de Threshold:**  
   Um valor de threshold para uma determinada imagem, passando o seu limiar e tipo de binarização que vai ser feita, podendo ser THRESH\_BINARY\_INV ou THRESH\_BINARY, que classifica a intensidade de pixel.

Se o peso do pixel maior que o primeiro limiar, para THRESH\_BINARY será branco e para THRESH\_BINARY será preto, abaixo disso basta realizar a inversão de lógica. O ajuste manual do limiar permite controlar a forma como as manchas internas e o contorno externo serão representados.

1. **4. Aplicação de Operações Morfológicas:**

As operações morfológicas atuam sobre a forma dos objetos na imagem binária, e o tamanho do kernel influencia na geração de ruídos e remoção de detalhes em imagens. Um kernel pequeno (3x3 ou 5x5) atua em detalhes locais da imagem, removendo ruídos pequenos, mas preservando detalhes sutis em um determinado objeto.Já um kernel grande (>10x10) atua em áreas maiores, ignorando e podendo até mesmo remover partes finais de um determinado objeto ou acabar unindo regiões próximas, ele é útil quando a imagem possui muitos ruídos.

1. **Detecção de Contornos:**  
   Após a binarização e morfologia, a função cv2.findContours é utilizada para extrair todos os contornos.
2. **Exibição dos Resultados:**  
   São exibidas as imagens intermediárias (como a imagem original, a imagem em escala de cinza, a imagem após o thresholding, a imagem processada morfologicamente e a imagem final com os contornos desenhados) para facilitar a validação e a análise de cada etapa do processamento.

Após aplicar esses processos nas imagens, ajustamos o threshold e implementamos uma forma de destaque de cada contorno para cada item solicitado na imagem, com o objetivo de representá-los da forma mais fiel possível ao objeto principal. Como resultado, obtivemos as seguintes imagens:





