Counting Objects in Images

Mateo Cardona Arias

Agosto 2025

Introducción

El conteo de objetos en imágenes digitales es una tarea fundamental en el área de la visión por computador, con aplicaciones que incluyen la gestión de inventarios, la monitorización del tráfico, el análisis ambiental y la investigación científica.

El presente trabajo se desarrolla siguiendo la guía "*Counting Objects in Images*" [1], utilizando la librería OpenCV en Python para implementar un flujo de procesamiento de imágenes capaz de detectar y contar objetos presentes en fotografías reales.

El objetivo general es diseñar y ejecutar un algoritmo que permita identificar y contabilizar objetos en imágenes capturadas en condiciones reales, evaluando la precisión del método en función del preprocesamiento, segmentación y detección de contornos.

Adicionalmente, se realizó una presentación en video para ilustrar el funcionamiento del proyecto. Dirección del video: `https://youtu.be/zC_FS1OL2D4`

Metodología

La metodología empleada sigue las fases sugeridas en la guía original y se adapta a las características de las imágenes utilizadas (fotografías de monedas, frutas y pilas tomadas personalmente).

Preprocesamiento

- Conversión de la imagen a escala de grises mediante `cv2.cvtColor`.
- Suavizado gaussiano (`cv2.GaussianBlur`) para eliminar ruido fino.

Segmentación

Debido a que las imágenes presentan sombras y variaciones de iluminación, se optó por utilizar el detector de bordes de Canny (`cv2.Canny`) en lugar de la umbralización de Otsu, ya que Canny es más robusto frente a cambios de luz.

Posteriormente, se aplicó una operación morfológica de `CLOSE` con un kernel de 9×9, con el fin de cerrar huecos y unir bordes cortados.

Detección de Objetos

Los contornos fueron extraídos con `cv2.findContours` y filtrados por área mínima (`cv2.contourArea >1000`) para descartar ruido y pequeñas imperfecciones del fondo.

Visualización y Conteo

Se generaron rectángulos delimitadores y etiquetas numéricas sobre cada objeto detectado utilizando `cv2.boundingRect`, `cv2.rectangle` y `cv2.putText`. Finalmente, los resultados fueron visualizados con `matplotlib`.

Código Implementado

El siguiente fragmento corresponde a la implementación funcional desarrollada:

```
imagen = cv2.imread("monedas.jpeg")
escalaGrises = cv2.cvtColor(imagen, cv2.COLOR_BGR2GRAY)
suavizado = cv2.GaussianBlur(escalaGrises, (7, 7), 0)
edges = cv2.Canny(suavizado, 50, 150)
kernel = np.ones((9, 9), np.uint8)
closed = cv2.morphologyEx(edges, cv2.MORPH_CLOSE, kernel)
contours, _ = cv2.findContours(closed, cv2.RETR_EXTERNAL,
                               cv2.CHAIN_APPROX_SIMPLE)
valid_contours = [c for c in contours if cv2.contourArea(c) > 1000]
```

Resultados

El algoritmo fue probado con fotografías propias de monedas, pilas y frutas sobre una superficie.

- En la imagen de monedas, el algoritmo detectó la mayoría de las monedas correctamente, aunque algunas pequeñas imperfecciones en la mesa aparecieron como falsos positivos.
- En la imagen de pilas, el resultado fue más preciso: las cuatro pilas fueron correctamente identificadas y contadas.
- En la imagen de las frutas, el resultado fue bastante preciso tambien: las tres frutas fueron identificadas correctamente con algunas imperfecciones.

Se observó que el filtrado por área es un factor determinante para descartar manchas pequeñas, y que el detector de Canny proporciona bordes definidos en condiciones de iluminación irregular.

Conclusiones

1. La metodología propuesta en la guía fue implementada con éxito, adaptándola al uso de Canny en lugar de Otsu para imágenes reales.

2. El uso de operaciones morfológicas fue esencial para mejorar la calidad de los bordes y garantizar un conteo más preciso.

3. Los resultados muestran que el método es efectivo para objetos de forma clara y contraste fuerte como por ejemplo con las pilas, pero requiere ajustes de parámetros en escenas más complejas como lo puede ser con las monedas y las frutas con brillos y sombras.

Referencias

Bibliografía

[1] Python for Research, *Counting Objects in Images*, Agosto 2025.