5. Segmentación de imágenes y caracterización de regiones.

Resumen

Segmentar una imagen en partes puede ser un paso hacia la comprensión de la escena. Si identificamos diferentes regiones, éstas pueden contarse, caracterizarse o clasificarse. Esta práctica de laboratorio cubre cómo realizar estas operaciones a un nivel introductorio utilizando imágenes simples capturadas en condiciones relativamente controladas.

Palabras clave Umbral mediante análisis de histograma • Algoritmo de Otsu • Etiquetado de componentes conectados • Caracterización de regiones

Contenido

1. Binarización

El enfoque de segmentación posiblemente más sencillo para segmentar una imagen es la binarización, es decir, asignar a cada píxel uno de dos valores posibles, normalmente objetos (o primer plano) y fondo. El método Otsu es un algoritmo clásico de umbralización , es decir, binarización de imágenes utilizando un umbral del nivel de gris. En el método de Otsu, este umbral se encuentra automáticamente.

E7 Veamos cómo utilizar y valorar este método. Considere la imagen monedas.pgm. Muestre su histograma y verifique que sea claramente bimodal. A partir de esa observación, elija manualmente un umbral (para ello utilice la variable myThreshold) y compare la segmentación utilizando este umbral con la de Otsu. Si el objetivo es extraer los objetos del fondo, ¿ cuál de los dos umbrales crees que es más adecuado para esta imagen?

Luego, a nivel de programación y usando nuestro código:

- ¿Qué función utilizamos para aplicar el método Otsu?
- ¿A qué módulo de Python pertenece?
- ¿Cómo binarizamos una imagen dado un umbral?
- ¿Qué valores tiene la imagen binarizada?

2. Etiquetado de regiones

E1 Si mostramos una imagen binarizada, podemos percibir diferentes “regiones” y la forma tradicional de identificarlas automáticamente es mediante un algoritmo de etiquetado de componentes conectados. ¿Qué función y módulo de Python utilizamos para ese propósito ? ¿Qué sucede y por qué si en esta llamada a función cambiamos de 0 a 1 como argumento para el fondo del parámetro?

Si presta atención a las regiones identificadas, notará que el resultado es de alguna manera indeseablemente ruidoso. Puedes ver que las regiones tienen muchos pequeños agujeros, y hay una región muy pequeña (¿qué tamaño tiene?).1 Casos simples como este pueden solucionarse mediante operaciones que pertenecen a un área conocida como morfología matemática. pero que no está cubierto en nuestro curso.

3. Caracterización de la región

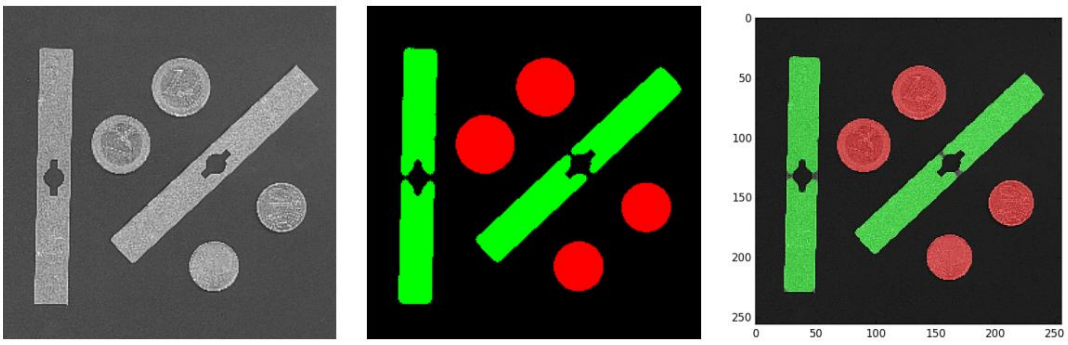
Finalmente, una vez que conocemos el conjunto de píxeles correspondientes a cada región, podemos calcular valores para caracterizarla: área, perímetro, cuadro delimitador, excentricidad, etc. Esas características pueden ser la base para otros cálculos y/o tomar algunas decisiones . . Por ejemplo, en nuestras imágenes de monedas podemos adivinar qué regiones E2 corresponden a monedas y cuáles otras son otra cosa. En un paso más, podemos distinguir diferentes tipos de monedas E3 y luego proceder directamente a contar el dinero. E4

4. Ejercicios

1. Intente encontrar una (buena) segmentación de las imágenes de las monedas. Para este y los siguientes ejercicios, puedes utilizar monedas, pero también monedas1 y monedas2. Sin embargo, las dos últimas imágenes son más desafiantes, en su mayoría sin utilizar morfología matemática. Así que no te preocupes si te resulta difícil conseguir buenos resultados; simplemente use esta experiencia para apreciar mejor cómo los problemas que son muy simples para los humanos pueden ser difíciles para las computadoras, y cómo las herramientas adicionales de procesamiento de imágenes y visión por computadora pueden ayudarnos en algunas situaciones. Nótese que, en este punto, todavía no podemos distinguir las monedas de otros objetos; sólo sabemos que hay diferentes regiones y su conjunto de píxeles.

   Nota al margen: observe que en situaciones más realistas el problema que estamos considerando es más desafiante porque nuestros algoritmos deberían funcionar para muchas más imágenes, no solo una o unas pocas, y las imágenes pueden ser incluso desconocidas o no estar disponibles en el momento.

[1] ¿Es este resultado subóptimo una consecuencia del algoritmo de etiquetado de componentes conectados?



(a) (b) (C)

Figura 1. Resultado de clasificar regiones en las clases “moneda” y “no-moneda” (Ejercicios 2): (a) la imagen original (monedas); (b) clase de moneda en rojo y clase sin moneda en verde; y (c) superposición del resultado de la clasificación en la imagen original

momento de desarrollo del algoritmo. Afortunadamente, en esos casos podemos incluir técnicas de aprendizaje automático (ML) para hacer nuestros sistemas más robustos y generales. Como sabes, hoy en día la visión por ordenador está íntimamente relacionada con el ML, hasta el punto de que cada vez es menos común desarrollar sistemas de visión por ordenador que no dependan del ML. Si tiene curiosidad sobre temas de investigación en visión por computadora, una buena fuente son las actas de las principales conferencias internacionales como CVPR, ICCV, ECCV, por ejemplo en The Computer Vision Foundation – Open Access.

2. Utilice el perímetro y el área de las regiones encontradas (o cualquier otra característica o combinación de características que considere útil o con la que quiera experimentar) para identificar aquellas que tienen una forma circular aproximada (que, en nuestro caso, corresponde a monedas).). Representa el resultado usando un color para la región circular y otro para aquellas que no lo son, similar a la Fig. 1 (Nota: usamos morfología matemática para obtener este resultado; está perfectamente bien si tu resultado no se ve así 1. Dado que labelConnectedComponents () devuelve tanto las imágenes binarias procesadas como las no procesadas y las regiones correspondientes, puede experimentar con ambas y ver si puede distinguir cuáles son las regiones de monedas en ambos casos.) Para cada imagen, cuente programáticamente y muestre (ya sea en la salida estándar o dibujada en la imagen) el número de monedas.

3. Utilizando el resultado del Ejercicio 2 como punto de partida, encuentre qué regiones corresponden a cada tipo de moneda (1 euro, 10 céntimos). La siguiente información puede resultar útil:

- Las imágenes de las monedas se tomaron con 50 píxeles por pulgada (1 pulgada = 2,54
- cm). los diámetros de las monedas de 1 euro y 10 céntimos son 23 mm y 19,5 mm respectivamente.

Opcionalmente, como en aquel ejercicio anterior, representar visualmente los dos tipos de monedas. Usando transparencia, superponga la imagen en color que representa cada moneda y la imagen original (de manera similar a la Fig. 1c), para facilitar el manual. comprobar si la clasificación de cada región es correcta y en qué medida coinciden las regiones reales y estimadas.

4. Por último, calcular automáticamente la cantidad total de dinero y, opcionalmente, mostrarlo como texto en la figura (como parte del título y/o como texto dibujado en la imagen). Compare los recuentos de dinero reales y estimados. Identifique los puntos fuertes y débiles de las diferentes partes de su solución. Puede pensar en posibles formas de mejorar la solución, pero no implementarlas.

5. (Opcional, de alguna manera avanzado o costoso) Desarrolle su propia implementación de (a) el método de Otsu y/o (b) el algoritmo de etiquetado de componentes conectados.

6. (Opcional) Pruebe el umbral multi-Otsu para el caso de imágenes con más de dos clases de objetos. Puede realizar pruebas con imágenes de monedas o con su propio conjunto de imágenes.

7. (Opcional, abierto) Existen otros métodos de segmentación además del de Otsu que también dependen del umbral. Elige y estudia un par de ellos. Realice algunas pruebas y compare esos algoritmos y los de Otsu en diferentes tipos de imágenes (monedas pe, textos, señales de tráfico). Analizar el comportamiento de los diferentes algoritmos bajo diferentes condiciones (características de la imagen, tipo de ruido, etc.)

8. (Opcional, avanzado) En lugar de segmentar una imagen a partir de píxeles individuales en regiones finales, una solución es sobresegmentar una imagen en los llamados “superpíxeles ”. Dado que los superpíxeles son algo entre píxeles y regiones, pueden usarse como una salida intermedia que puede beneficiar problemas posteriores. Por ejemplo, se puede mejorar la eficiencia y el rendimiento de los algoritmos de segmentación. Uno de esos métodos de superpíxeles es SLIC. (agrupación iterativa lineal simple). Aprenda sobre esto y luego pruébelo en nuestras imágenes de monedas u otras imágenes. Puedes usar slic() de skimage.