

Procesamiento Digital de Imágenes

Procesamiento morfológico

Noelia Revollo

Facultad de Ingeniería– UNJU – CONICET

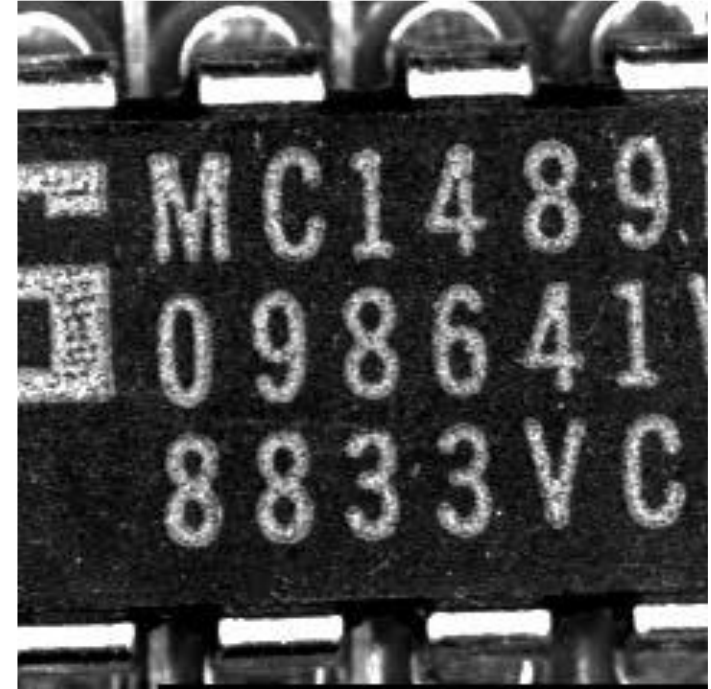
grevollo@fi.unju.edu.ar



PDI – Limitaciones del procesamiento lineal

Imaginemos por un momento el siguiente problema: tenemos una imagen como la de la derecha, y necesitamos segmentar los bordes de las letras para un ulterior paso de reconocimiento (por ejemplo, en una línea de montaje robótica, una cámara necesita reconocer los componentes para decidir un paso u otro de ensamblado).

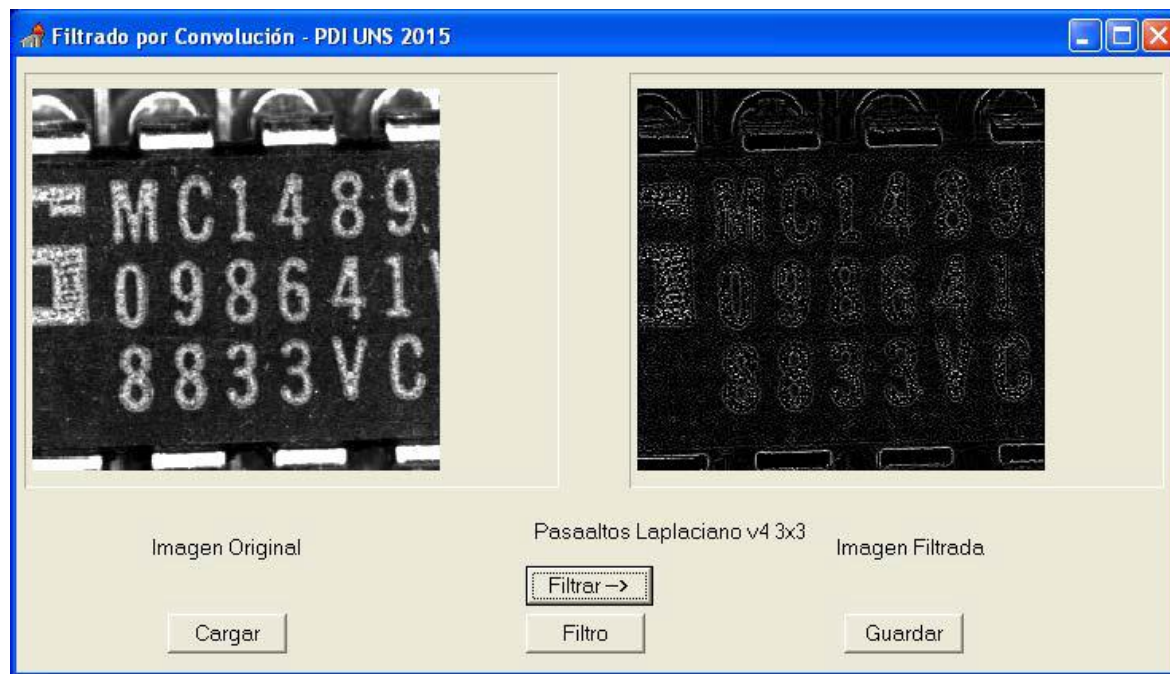
Como ya vimos, el filtrado Laplaciano es adecuado para detectar los bordes de los objetos, por lo que podemos intentar aplicarlo en este caso.



PDI – Limitaciones del procesamiento lineal

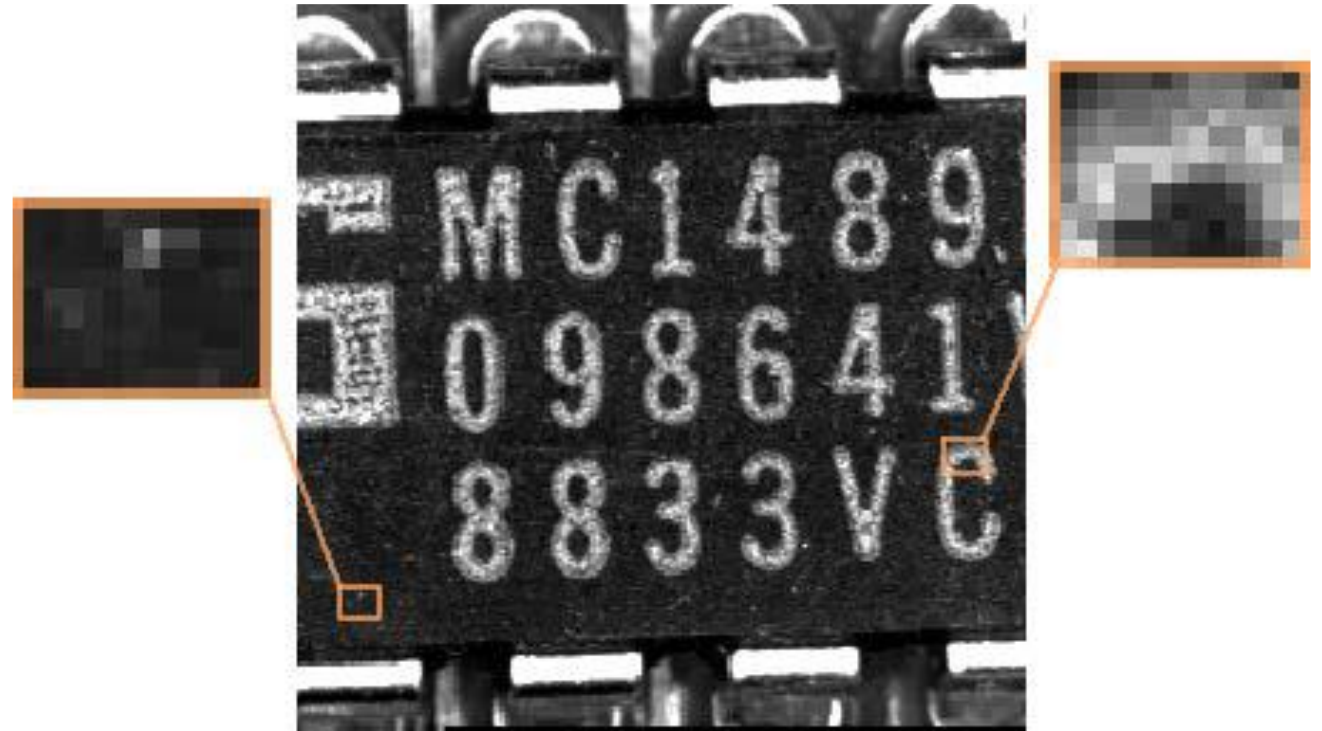
Sin embargo, si utilizamos este u otros filtros similares, el resultado será invariablemente insatisfactorio, como muestra lo que genera nuestra implementación de referencia.

Por qué ocurre esto?



PDI – Limitaciones del procesamiento lineal

Viendo detalles de la imagen con mayor detalle vemos que la parte blanca (figura) no es muy pareja. Lo mismo ocurre con el fondo. Esas diferencias locales de luminancia son amplificadas por el Laplaciano o cualquier filtro similar.

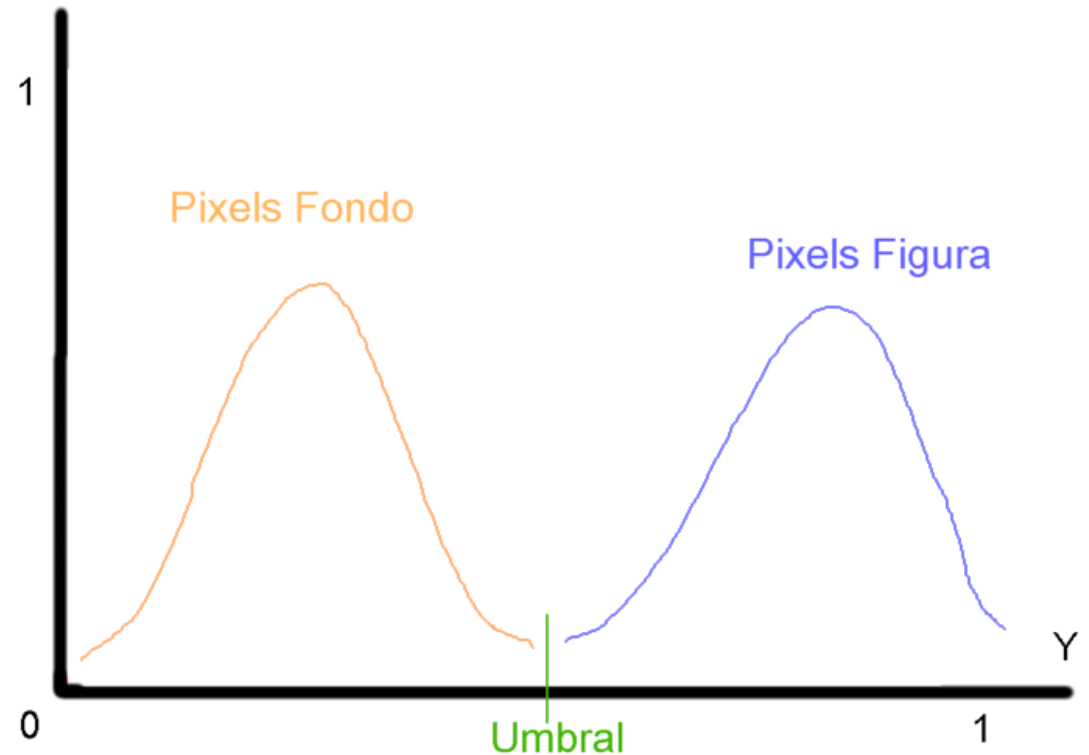


PDI – Limitaciones del procesamiento lineal

Una posible solución sería *binarizar* la imagen, es decir, transformarla a pixels con luminancia 1 o 0.

Esto se logra *coercionando* todos los pixels a 1 o 0 dependiendo de un valor *umbral* de luminancia.

En la figura vemos un histograma idealizado donde el umbral separa adecuadamente las luminancias.



PDI – Limitaciones del procesamiento lineal

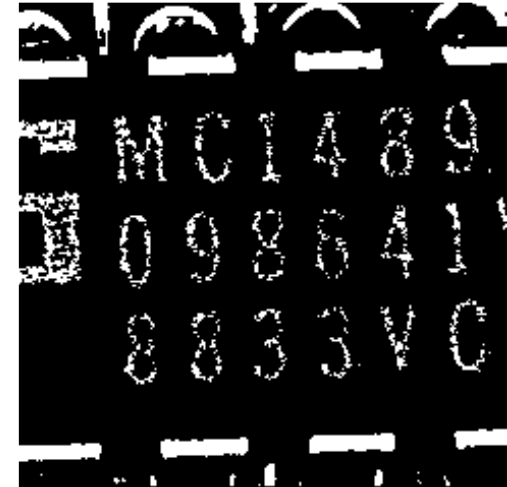
Pero qué ocurriría si el histograma no muestra claramente una separación entre la distribución de luminancias de los pixels figura y los pixels fondo?

Un umbralizado en algún valor posible tendrá seguramente errores (pixels blancos que deberían haber sido negros o viceversa).



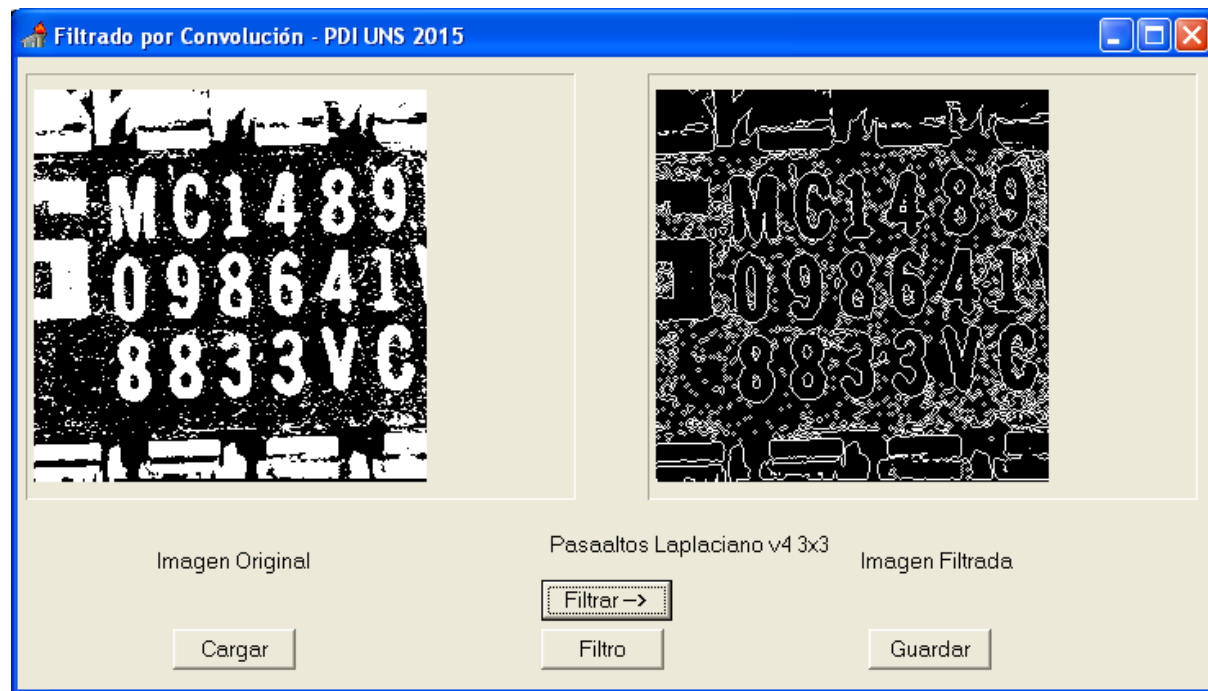
PDI – Limitaciones del procesamiento lineal

Vemos aquí dos casos posibles. En el primero el umbral fue demasiado bajo. Todos los pixels figura son blancos, pero algunos pixels fondo también lo son. En el segundo caso el umbral es demasiado alto. Todos los pixels fondo son negro, pero algunos pixel figura también son negros.



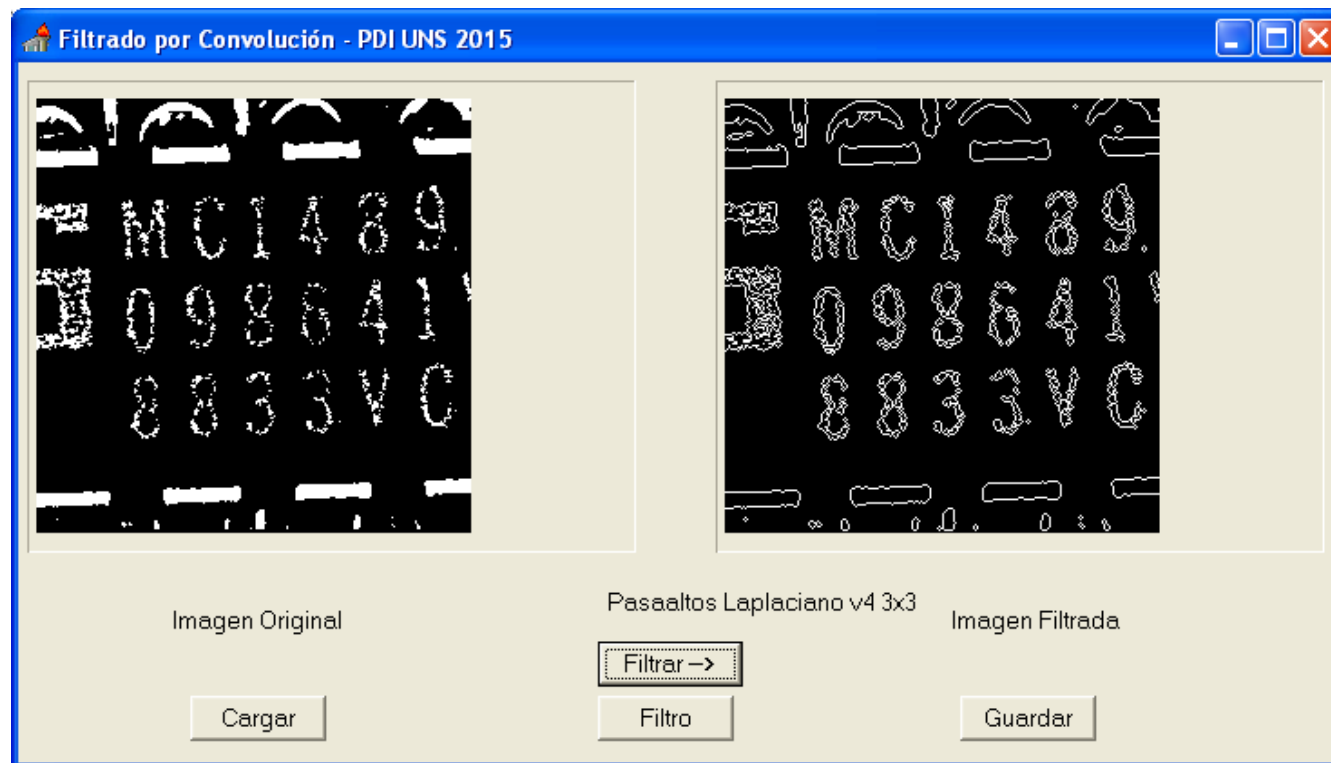
PDI – Limitaciones del procesamiento lineal

Al aplicar Laplaciano a la primera umbralización, se ve una buena segmentación de los bordes de las letras, pero se observa también la aparición de un sinnúmero de pequeñas segmentaciones de *ruido* en el fondo.



PDI – Limitaciones del procesamiento lineal

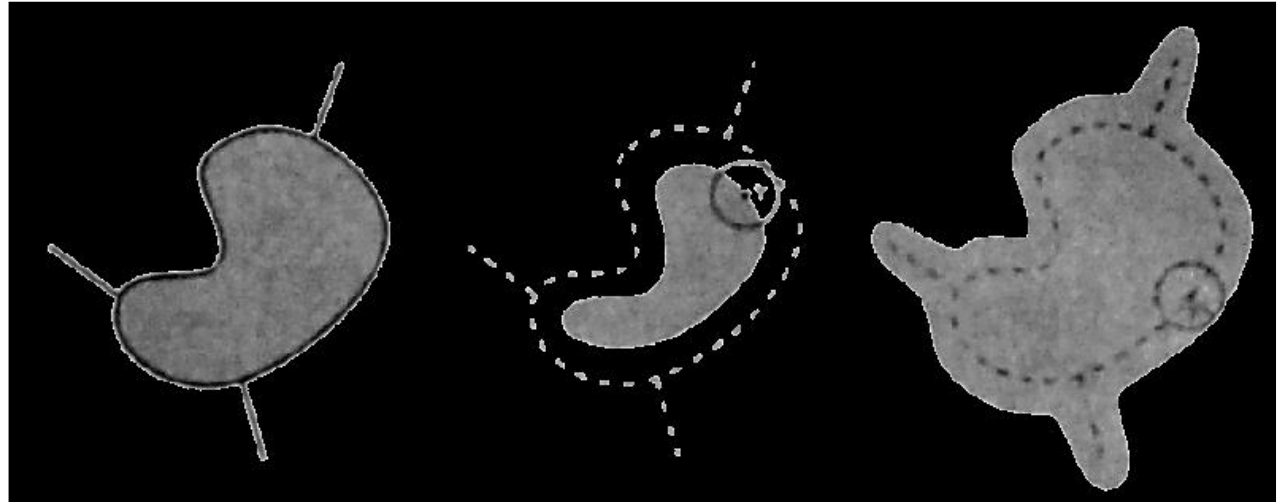
En el segundo caso, si bien desaparece el ruido en el fondo, la segmentación de las letras está fragmentada o cortada en algunos casos, lo cual puede generar problemas a la hora de aplicar algún tipo de reconocimiento.



PDI – Procesamiento morfológico

Para solucionar este y otros tipos importantes de problemas es que surge el *procesamiento morfológico*.

Las operaciones básicas de la morfología binaria (en blanco o negro) son la *erosión* y la *dilatación*.



En la figura vemos una figura de ejemplo (izq.) y el resultado de aplicarle erosión y dilatación.

https://es.wikipedia.org/wiki/Morfología_matemática

PDI – Procesamiento morfológico binario

La descripción matemática se basa en definir un *elemento estructurante* (en la figura anterior, un disco), el cual tiene un *punto de inserción* (en la figura anterior, el centro del disco).

De esa manera, la erosión se obtiene al insertar el elemento estructurante en todo punto fondo de la imagen y borrando (convirtiendo en fondo) todo punto tocado por el mismo.

La dilatación, recíprocamente, se obtiene al insertar el elemento estructurante en todo punto figura, y convirtiendo en figura todo punto tocado por el mismo.



PDI – Procesamiento morfológico binario

Si bien hay un gran número de casos donde es útil definir elementos estructurantes diferentes, por ahora nos concentraremos en un caso básico en el cual el elemento estructurante es un cuadrado de 3x3 pixels, cuyo punto de inserción es el pixel central.

De esa manera, para erosionar una imagen, la recorreremos buscando sus pixels fondo (con luminancia cero), y convirtiendo en cero los pixels vecinos, y reciprocamente para dilatar.

Este procesamiento, entonces, termina siendo computacionalmente similar a la convolución. Al erosionar, si alguno de los pixels es fondo, el pixel final es fondo, y a la inversa al dilatar. De todas maneras, tener en cuenta que se trata de operaciones no lineales.



PDI – Procesamiento morfológico binario

Suponemos que A es el objeto o figura a procesar (foreground = 1 = blanco) y B el elemento estructurante utilizado. Tanto A como B pueden representarse como conjuntos, donde A es el conjunto de pixels de la figura, y B el conjunto de “exels” (element-cells):

$$A = \{ (x,y) \mid I(x,y) = 1 \}$$

% el conjunto de pixels “prendidos” en la imagen I

$$B = \{ (-1,-1), (-1,0), \dots, (1,0), (1,1) \}$$

% elemento estructurante 3x3 centrado en (0,0)

Valen las operaciones conjuntistas usuales de unión, intersección y complemento.

Definimos también una operación de traslación: B_z representa todos los pixels de B trasladados por el vector z.



PDI – Procesamiento morfológico binario

De esa forma la dilatación de una imagen que contiene la figura A por el elemento estructurante B puede expresarse como la unión de todos los elementos B, cada uno de ellos trasladado a los pixels de la imagen A:

$$A \oplus B = \bigcup_{a \in A} B_a$$

La dilatación es conmutativa, por lo que vale

$$A \oplus B = B \oplus A = \bigcup_{b \in B} A_b$$



PDI – Procesamiento morfológico binario

Aplicar dilatación al ejemplo de binarización que vimos (con muchos falsos negativos) produce un resultado que es adecuado para luego aplicar un operador Laplaciano y reconocer los caracteres.



PDI – Procesamiento morfológico binario

La erosión es la operación dual de la binarización. De hecho, si la figura está en negro sobre fondo blanco, la dilatación tendrá el efecto de erosionar la imagen:

$$A \ominus B = \overline{\overline{A} \oplus \overline{B}}$$

Podemos formalizar la erosión de otras maneras, además:

$$A \ominus B = A / \bigcup_{a \in \overline{A}} B_a = \bigcap_{b \in B} A_b = \{x | B_x \subseteq A\}$$



PDI – Procesamiento morfológico binario

En el caso de la imagen binarizada con un umbral bajo (muchos falsos positivos), vemos que la erosión produce un efecto también adecuado.

