

Universidad de Guanajuato

DIVISIÓN DE CIENCIAS E INGENIERÍAS

CAMPUS LEÓN

Fundamentos de Procesamiento Digital de Imágenes

Grupo:A

Tarea 12 - Operaciones Morfológicas Unidad 2

FECHA DE ENTREGA: 15 DE NOVIEMBRE DE 2022

Alumno:

Miguel Ángel Hernández Tapia

Profesor:

Dr. Arturo González Vega

Índice

1. Introducción	3
2. Metodología	5
3. Resultados	6
4. Discusiones	12
5. Conclusión	12
A. Código	13

Fundamentos de Procesamiento Digital de Imágenes

Tarea 12 - Operaciones Morfológicas

Miguel Ángel Hernández Tapia

15 de noviembre de 2022

Resumen

En esta práctica se revisan las operaciones morfológicas más importantes, donde estas dependen de la forma de elemento. Se realizaron las combinaciones necesarias para cumplir con los objetivos y también el etiquetado de elementos en una imagen, así como la obtención de sus características.

1. Introducción

El lenguaje de la morfología matemática es la teoría de conjuntos. Como tal, la morfología ofrece un enfoque unificado y poderoso para numerosos problemas de procesamiento de imágenes. Cuando estamos trabajando con imágenes, los conjuntos en morfología matemática representan objetos como imágenes. Las operaciones morfológicas se definen en términos de conjuntos. En el procesamiento de imágenes, utilizamos morfología con dos tipos de conjuntos de píxeles: objetos y elementos estructurales (SE).

Las transformaciones morfológicas son algunas operaciones simples basadas en la forma de la imagen. Normalmente se realiza en imágenes binarias. Necesita dos entradas, una es nuestra imagen original, la segunda se llama elemento estructural o kernel que decide la naturaleza de la operación.

Erosión

La idea básica de la erosión es como la erosión del suelo solamente, erosiona los límites del objeto en primer plano (siempre trate de mantener el primer plano en blanco).

Dilatación

Es justo lo contrario de la erosión. Por lo tanto, aumenta la región blanca en la imagen o aumenta el tamaño del objeto en primer plano.

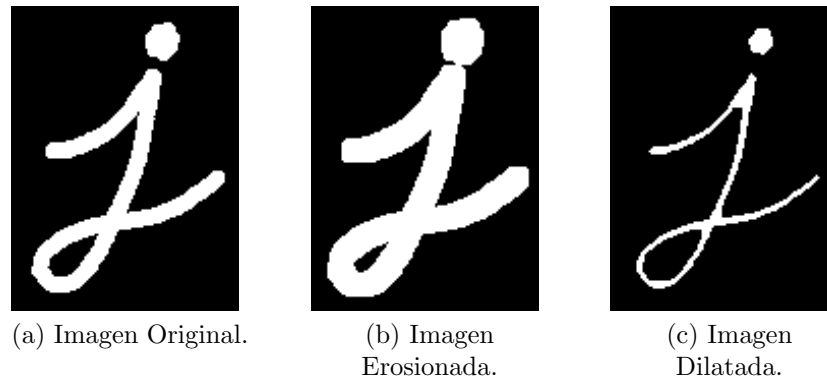


Figura 1: Resultado de la erosión y dilatación (Comparación).

Apertura

La apertura es solo otro nombre de erosión seguida de dilatación. Es útil para eliminar el ruido.

Cerradura

El cierre es inverso de la apertura, la dilatación seguida de la erosión. Es útil para cerrar pequeños agujeros dentro de los objetos en primer plano, o pequeños puntos negros en el objeto.



Figura 2: Resultado de la apertura y cerradura (Comparación).

El elemento estructural es la forma en cómo se decide hacer la operación, ya sea aumentar esquinas, la parte de arriba, abajo a los lados o con formas de todo tipo y esto se define por el kernel, donde hay 1's se añade o quita según la operación.[1] [2] [3]

$$\begin{pmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{pmatrix} \quad \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} \quad \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix}$$

2. Metodología

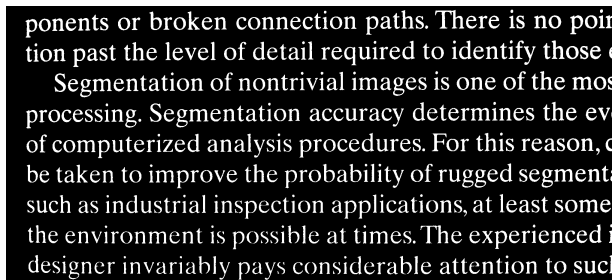
Este trabajo se desarrollo en Google Colab. En este ambiente se puede compilar el lenguaje Python con una basta cantidad de librerías ya incluidas sin necesidad de instalarlas. Se utilizaron las librerías cv2 (OpenCV), numpy y matplotlib.pyplot. La primera sirve para manejo de imágenes como objetos que tiene diversos métodos y funciones definidas. Numpy es una librería optimizada para operaciones de arreglos (vectores, matrices, etc.), muy útil para reducir y optimizar las operaciones con sus métodos. Por último, matplotlib.pyplot es una función para graficar, funciona muy similar al plot de Matlab y Plotly para tener una interfaz más completa y visualizar con acercamiento a las imágenes en notebooks.

Los pasos son:

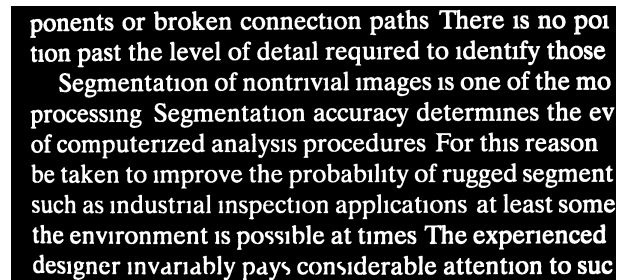
1. Leer la imagen *Fig1022(a).tif* y hacer las operaciones morfológicas necesarias para contar la cantidad de letras lo mejor posible.
2. Haga un programa que cuente el número de rectángulos que aparecen en una imagen. Genere una tabla donde parezca la siguiente información de cada rectángulo encontrado: área del rectángulo, coordenadas del centro de gravedad, y las coordenadas de cada vértice. Pruebe el programa con la imagen *Fig1013(a).tif*.
3. Para la imagen *Fig1026(a).tif* utilice una umbralización para detectar los objetos cilíndricos, cuéntelos, etiquételos, mida su área, coordenadas del centroíde y promedio de intensidad.

3. Resultados

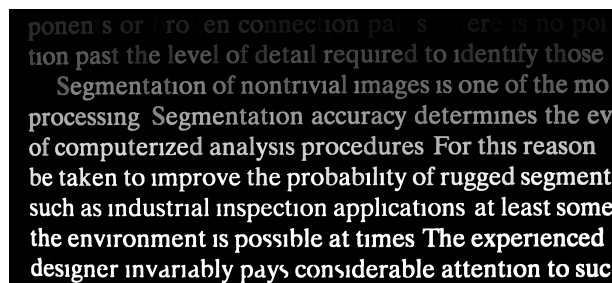
1.- El resultado del conteo de letras es 408 se realizó una dilatación de un pixel en todo el borde de las letras para la parte superior y a partir del pixel 700 hasta abajo se realizó una dilatación de 2 píxeles. También se hizo una apertura para reducir las ies y separar algunas letras. Se hizo un primer etiquetado con el cual se ubicaron las áreas pequeñas y las que chocan en el lado derecho y finalmente, volver a etiquetar el resultado. El proceso se muestra en la Figura 3.



(a) Imagen Original.



(b) Imagen Limpia.



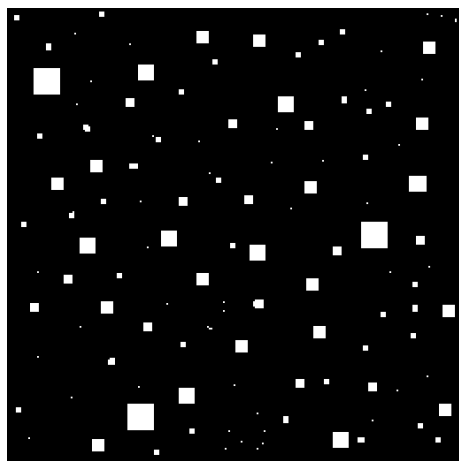
(c) Imagen Etiquetada.

Figura 3: Proceso para conteo de letras.

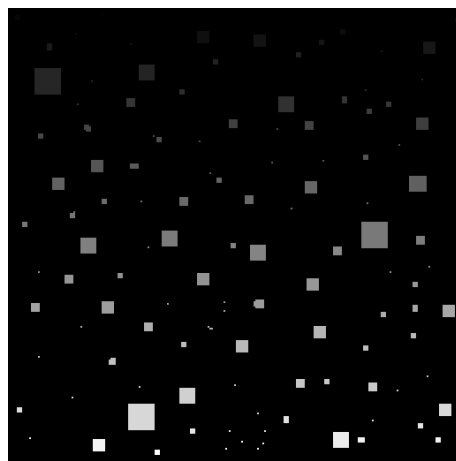
2.- Se utilizó la función *connectedComponentsWithStats* de OpenCV para obtener el número de elementos, la matriz con los elementos etiquetados, el área, vértices, dimensiones y centroide. La tabla con las características de cada rectángulo se muestran en la Tabla 1. La imagen original y la etiquetada se muestra en la Figura 4.

3.- Se realizó una umbralización binaria con la función *threshold* de OpenCV de 135 – 245 para la parte superior y del pixel 350 hacía abajo se realizó una umbralización de 110 – 240. El proceso se muestra en la Figura 5.

Luego se hizo una apertura para borrar los pequeños píxeles blancos y finalmente, se binarizó de 0 – 255 para etiquetar cada elemento. Por último, se calculo la intensidad media de cada elemento de la primera imagen como se muestra en la Figura 6 y se imprimieron los resultados en la Tabla 2.



(a) Imagen Original.

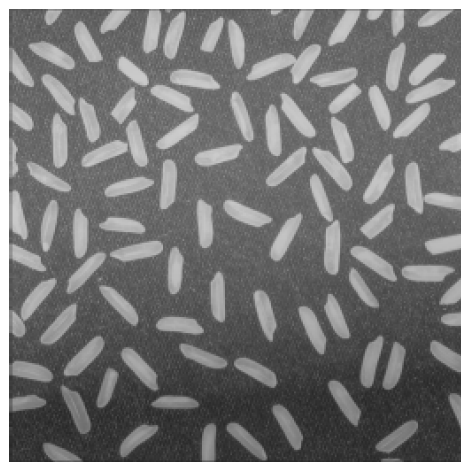


(b) Imagen Etiquetada.

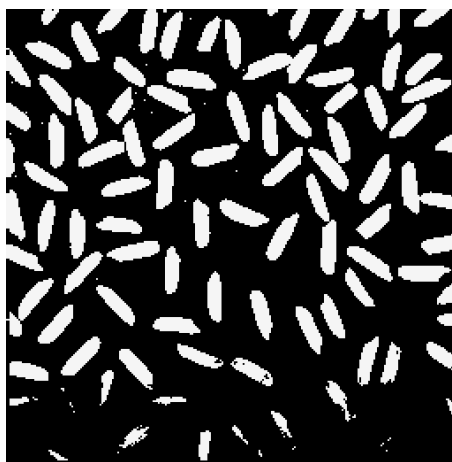
Figura 4: Etiquetado de cada rectángulo.

Etiqueta	Centroíde	Área	Vértices
0	[160. 10.]	81	[156 6] [165 6] [156 15] [165 15]
1	[712. 10.]	9	[711 9] [714 9] [711 12] [714 12]
2	[16. 16.]	81	[12 12] [21 12] [12 21] [21 21]
3	[736. 13.]	9	[735 12] [738 12] [735 15] [738 15]
4	[760. 20.5]	18	[759 18] [762 18] [759 24] [762 24]
5	[568. 40.]	81	[564 36] [573 36] [564 45] [573 45]
6	[331. 49.]	441	[321 39] [342 39] [321 60] [342 60]
7	[115. 43.]	9	[114 42] [117 42] [114 45] [117 45]
8	[427. 55.]	441	[417 45] [438 45] [417 66] [438 66]
9	[532. 58.]	81	[528 54] [537 54] [528 63] [537 63]
10	[715. 67.]	441	[705 57] [726 57] [705 78] [726 78]
11	[70. 65.5]	108	[66 60] [75 60] [66 72] [75 72]
12	[208. 61.]	9	[207 60] [210 60] [207 63] [210 63]
13	[634. 73.]	9	[633 72] [636 72] [633 75] [636 75]
14	[493. 79.]	81	[489 75] [498 75] [489 84] [498 84]
15	[352. 91.]	81	[348 87] [357 87] [348 96] [357 96]
16	[235. 109.]	729	[222 96] [249 96] [222 123] [249 123]
17	[67. 124.]	2025	[45 102] [90 102] [45 147] [90 147]
18	[703. 121.]	9	[702 120] [705 120] [702 123] [705 123]
19	[142. 124.]	9	[141 123] [144 123] [141 126] [144 126]
20	[295. 142.]	81	[291 138] [300 138] [291 147] [300 147]
21	[607. 139.]	9	[606 138] [609 138] [606 141] [609 141]
22	[472. 163.]	729	[459 150] [486 150] [459 177] [486 177]
23	[571. 155.5]	108	[567 150] [576 150] [567 162] [576 162]
24	[208. 160.]	225	[201 153] [216 153] [201 168] [216 168]
25	[646. 163.]	81	[642 159] [651 159] [642 168] [651 168]
26	[118. 163.]	9	[117 162] [120 162] [117 165] [120 165]
27	[613. 175.]	81	[609 171] [618 171] [609 180] [618 180]
28	[703. 196.]	441	[693 186] [714 186] [693 207] [714 207]
29	[382. 196.]	225	[375 189] [390 189] [375 204] [390 204]
30	[511. 199.]	225	[504 192] [519 192] [504 207] [519 207]
31	[134.5 203.5]	126	[129 198] [141 198] [129 210] [141 210]
32	[457. 205.]	9	[456 204] [459 204] [456 207] [459 207]
33	[55. 217.]	81	[51 213] [60 213] [51 222] [60 222]
34	[247. 217.]	9	[246 216] [249 216] [246 219] [249 219]
35	[256. 223.]	81	[252 219] [261 219] [252 228] [261 228]

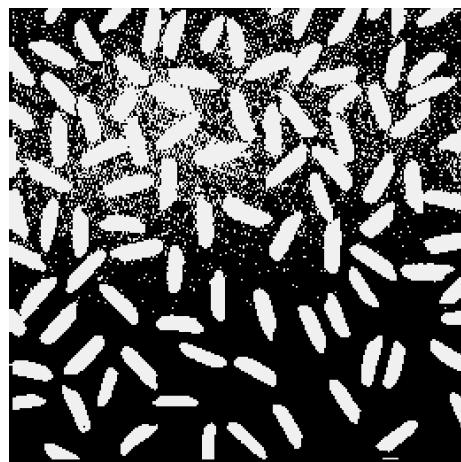
Tabla 1: La tabla loca weon



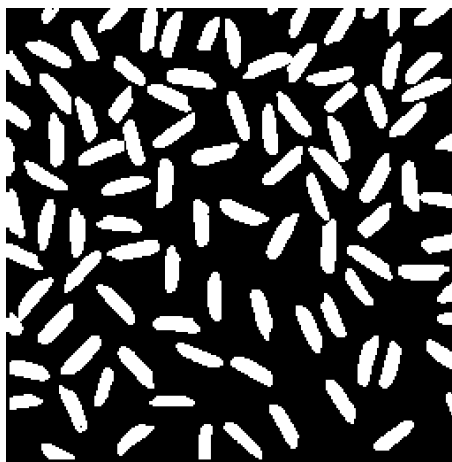
(a) Imagen Original.



(b) Umbral Superior.

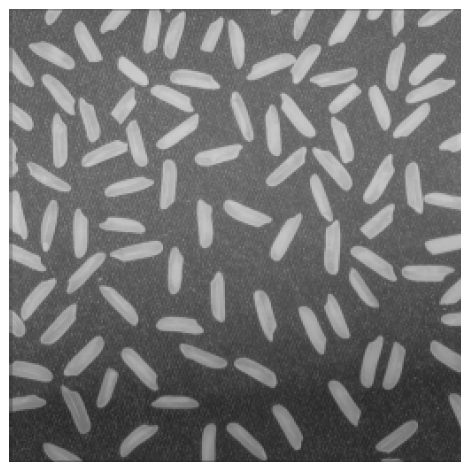


(c) Umbral Inferior.

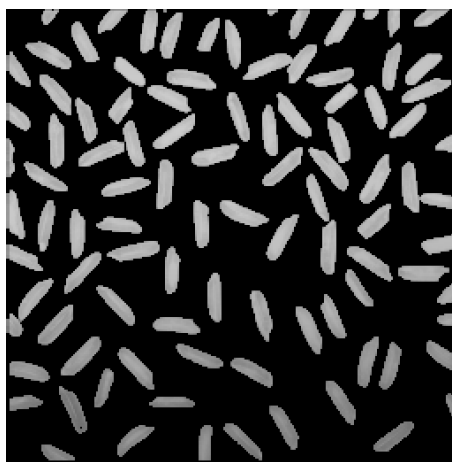


(d) Imagen Limpia.

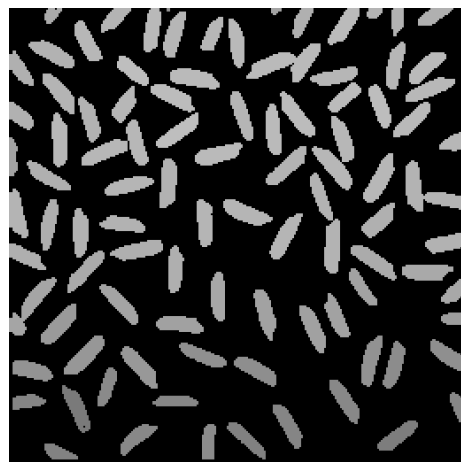
Figura 5: Proceso de limpieza de la imagen de las células.



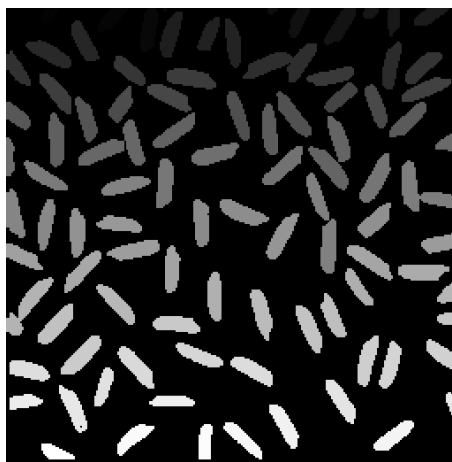
(a) Imagen Original.



(b) Intensidades de la Original.



(c) Intensidades Media.



(d) Imagen Etiquetadas.

Figura 6: Intensidades medias y etiquetadas.

Etiqueta	Centroíde	Área	Intensidad Media
0	[48.1 7.3]	240	169.56666666666666
1	[118.8 11.7]	516	179.6046511627907
2	[161.3 24.1]	756	184.7883597883598
3	[179.5 1.5]	16	146.0
4	[300.8 2.7]	72	164.27777777777777
5	[330.4 14.8]	476	176.2689075630252
6	[378.3 19.8]	756	186.88359788359787
7	[412.1 5.1]	180	180.44444444444446
8	[438.3 13.3]	372	177.89247311827958
9	[478.9 8.8]	484	175.69421487603304
10	[186.2 29.9]	812	181.50738916256157
11	[229.4 29.8]	568	183.19718309859155
12	[256.7 39.2]	744	185.19354838709677
13	[87.8 38.1]	776	182.7164948453608
14	[49. 51.2]	852	178.36150234741785
15	[435.8 65.]	772	181.57512953367876
16	[313.5 62.1]	1668	185.9640287769784
17	[12.1 68.9]	576	175.74305555555554
18	[471. 66.2]	772	186.25906735751295
19	[139.2 70.6]	636	190.46540880503144
20	[368.2 77.]	744	182.1451612903226
21	[208.6 79.2]	924	185.5021645021645
22	[56.9 95.3]	820	180.26341463414633
23	[474.8 91.4]	820	183.7658536585366
24	[378.7 100.8]	612	186.8562091503268
25	[185. 99.9]	828	188.12077294685992
26	[418.8 110.3]	740	188.54594594594596
27	[130.2 109.6]	636	183.11949685534591
28	[262.7 120.9]	792	181.03535353535352
29	[324.7 120.9]	888	188.0810810810811
30	[90.7 125.5]	744	178.05913978494624
31	[12.5 122.1]	536	177.03731343283582
32	[455.3 126.6]	768	185.43229166666666
33	[298.2 137.8]	832	185.92307692307693
34	[56.6 149.2]	896	183.16517857142858
35	[191.2 139.6]	812	189.95566502463055

Tabla 2: La tabla loca weon

4. Discusiones

1.- Se hicieron las operaciones necesarias para expandir las letras y unir las letras "sz ." de la parte de abajo y se hizo una apertura para eliminar lo que no se alcanzó a unir con las palabras. Por último, se etiquetó lo resultante para hacer ciclos donde se condiciona si una letra choca con el borde de la derecha se borra esa letra y además, si el área es bajo un umbral de tamaño se borra, para eliminar los puntos de las "i", puntos y comas. El resultado se vuelve a etiquetar para obtener el número de elementos restantes que debería coincidir con las letras totales. Haciendo acercamiento en la imagen limpia, se nota que se juntaron 2 pares de letras y una "." que quedó separada. Con esto a tomar en cuenta, el resultado verdadero variaría en 1 carácter

2.- Se diseñó una función especial para detectar rectángulos pero con el problema de detectar otras figuras o incluso rectángulos de 1 ó 2 píxeles no se pueden detectar correctamente. Con la función de OpenCV se hace sin problema con todo tipo de figura calculando incluso, el área, vértices, dimensiones y centroides y con una velocidad hasta 18 veces más rápido. Por lo que resulta por creces más conveniente que intentar una función con los métodos de Python. Los centroides tienden hacia arriba y hacia la izquierda como preferencia cuando el rectángulo no tiene un centro definido, o sea, se redondea al mínimo en estos casos.

3.- Ya que la imagen original tiene un difuminado que va de claro a oscuro de arriba a abajo, por lo que, se requirió de 2 umbralizados para cada mitad, ya que uno leve carcome a uno y el otro satura y viceversa. Uniendo ambos umbralizados, se realiza una apertura para borrar los puntos blancos suspendidos. Luego se etiqueta pero para obtener la intensidad media de cada célula, se crea una máscara para filtrar la imagen limpia con la original y obtener las intensidades originales, se calcula la media y se imprimen estos valores en la última tabla.

5. Conclusión

Las operaciones morfológicas son muy útiles para poder obtener configuraciones donde se pueden hacer mejor análisis con combinaciones de las mismas.

El etiquetado de elementos es muy útil para analizar las figuras, contar, obtener detalles e incluso, borrar según se cumpla alguna condición para ello. El método para etiquetar de OpenCV es bastante veloz para lo que se calcula, no solo la imagen (matriz) con etiquetas, sino también sus características.

La umbralización es un proceso que incluso se puede hacer en mitades con imágenes con degradados, esto ya que, generalmente las umbrales son incompatibles en estos casos. Cuando la imagen es uniforme, se puede con una sola umbralización.

Finalmente, Las intensidades medias también están afectadas por el degradado de la imagen

original, por lo que pareciera con una especie de etiquetado pero no lo es.

A. Código

El código diseñado para esta práctica se encuentra en un repositorio de Github para su visualización y con la opción de abrir en Google Colab.

Referencias

- [1] Chung, B. W. (2017). *Getting Started with Processing and OpenCV*, pages 1–37. Apress, Berkeley, CA.
- [2] Solem, J. (2012). *Programming Computer Vision with Python: Tools and algorithms for analyzing images*, pages 73–273. O'Reilly Media.
- [3] Woods, G. . (2018). *Digital Image Processing*, pages 1–200. Pearson, New York, NY.