

La traducción de esta página aún no se ha actualizado a la versión más reciente. Haga clic aquí para ver la última versión en inglés.

imsegkmeans

Segmentación de imágenes basada en la agrupación K-medias

Sintaxis

```
L = imsegkmeans(I,k)
[L,centers] = imsegkmeans(I,k)
L = imsegkmeans(I,k,Name,Value)
```

Descripción

`L = imsegkmeans(I,k)` segmenta la imagen `I` en `k` grupos realizando una agrupación K-medias y devuelve la salida etiquetada segmentada en `L`. ejemplo

`[L,centers] = imsegkmeans(I,k)` también devuelve las ubicaciones de los centroides del grupo `centers`. ejemplo

`L = imsegkmeans(I,k,Name,Value)` utiliza argumentos de nombre-valor para controlar aspectos del algoritmo de la agrupación K-medias.

Ejemplos

[contraer todo](#)

✓ Segmentar imagen en escala de grises utilizando la agrupación K-medias

Lea una imagen en el área de trabajo.

Probar este ejemplo

 **Copy Command**

```
I = imread("cameraman.tif");
imshow(I)
title("Original Image")
```

 Get ▼

Original Image



Segmente la imagen en tres regiones utilizando la agrupación K-medias.

```
[L,Centers] = imsegkmeans(I,3);  
B = labeloverlay(I,L);  
imshow(B)  
title("Labeled Image")
```

 Get ▼

Labeled Image




▼ Mejorar la segmentación K-medias utilizando información espacial y de textura

Lea una imagen en el área de trabajo. Reduzca el tamaño de la imagen para que el ejemplo se ejecute más rápidamente.

Probar este ejemplo

 [Copy Command](#)

```
RGB = imread("kobi.png");  
RGB = imresize(RGB,0.5);  
imshow(RGB)
```

 Get ▼



Segmente la imagen en dos regiones utilizando la agrupación K-medias.

```
L = imsegkmeans(RGB,2);  
B = labeloverlay(RGB,L);  
imshow(B)  
title("Labeled Image")
```

 Get ▼

Labeled Image



Varios píxeles están mal etiquetados. El resto del ejemplo muestra cómo mejorar la segmentación K-medias complementando la información sobre cada píxel.

Complemente la imagen con información sobre la textura en el entorno de cada píxel. Para obtener la información sobre la textura, filtre una versión en escala de grises de la imagen con un conjunto de filtros de Gabor.

Cree un conjunto de 24 filtros de Gabor, que abarquen 6 longitudes de onda y 4 orientaciones.

```
wavelength = 2.^(0:5) * 3;  
orientation = 0:45:135;  
g = gabor(wavelength,orientation);
```

 Get ▼

Convierta la imagen a escala de grises.

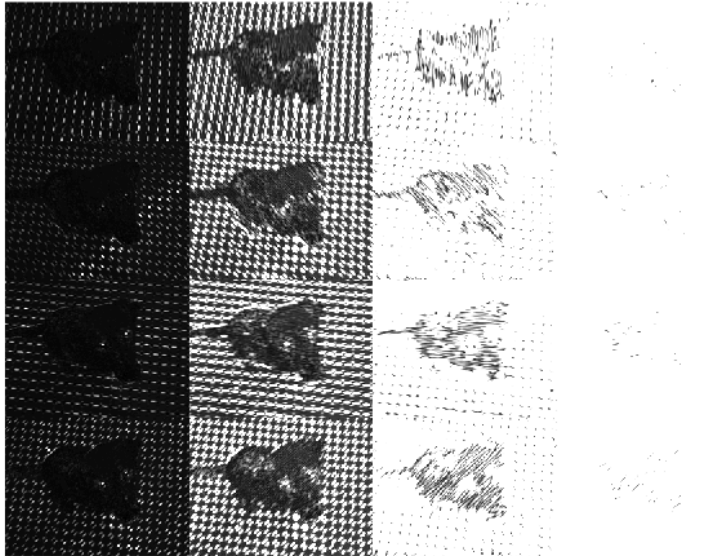
```
I = im2gray(im2single(RGB));
```

 Get ▼

Filtre la imagen en escala de grises utilizando los filtros de Gabor. Muestre la imagen filtrada con 24 filtros en un montaje.

```
gabormag = imgaborfilt(I,g);  
montage(gabormag,"Size",[4 6])
```

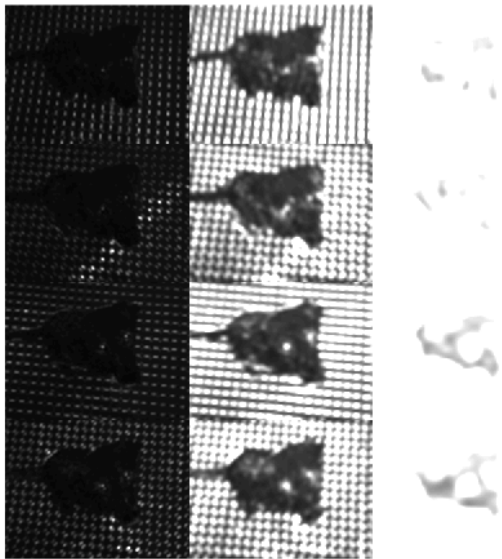
 Get ▼



Suavice cada imagen filtrada para eliminar las variaciones locales. Muestre las imágenes suavizadas en un montaje.

```
for i = 1:length(g)
    sigma = 0.5*g(i).Wavelength;
    gabormag(:, :, i) = imgaussfilt(gabormag(:, :, i), 3*sigma);
end
montage(gabormag, "Size", [4 6])
```

Get ▼



Complemente la información sobre cada píxel con información sobre la ubicación espacial. Esta información adicional permite que el algoritmo de agrupación K-medias prefiera las agrupaciones que estén más cerca entre ellas espacialmente.

Obtenga las coordenadas x e y de todos los píxeles de la imagen de entrada.

```
nrows = size(RGB,1);
ncols = size(RGB,2);
[X,Y] = meshgrid(1:ncols,1:nrows);
```

Get ▼

Concatene la información sobre la intensidad, la información sobre la textura del entorno y la información espacial sobre cada píxel.

En este ejemplo, el conjunto de características incluye la imagen de intensidad I en lugar de la imagen de color original RGB. La información sobre el color se omite en el conjunto de características dado que el color amarillo del pelo del perro es similar al tono amarillo de los azulejos. Los canales del color no proporcionan suficiente información distintiva sobre el perro y el entorno como para hacer una segmentación limpia.

```
featureSet = cat(3,I,gabormag,X,Y);
```

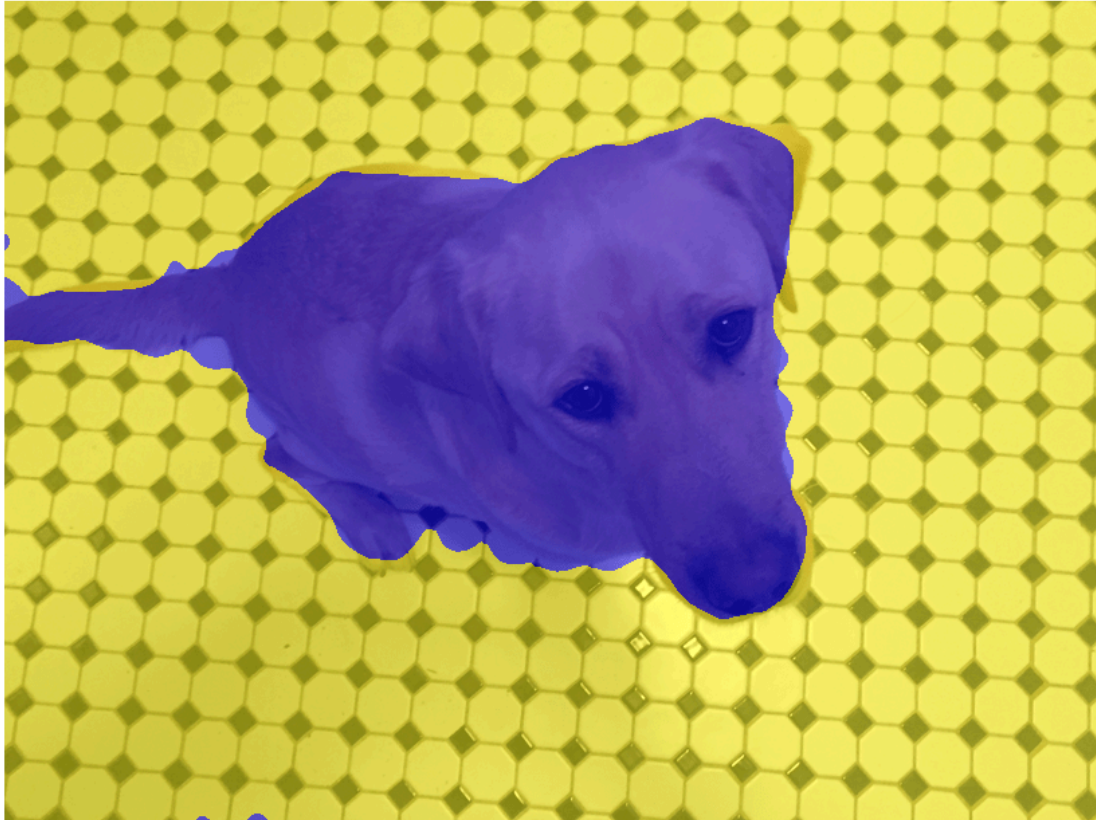
 Get ▾

Segmente la imagen en dos regiones utilizando la agrupación K-medias con el conjunto de características complementario.

```
L2 = imsegkmeans(featureSet,2,"NormalizeInput",true);  
C = labeloverlay(RGB,L2);  
imshow(C)  
title("Labeled Image with Additional Pixel Information")
```

 Get ▾

Labeled Image with Additional Pixel Information



✓ Comprimir una imagen de color utilizando la segmentación K-medias

Lea una imagen en el espacio de trabajo.

Probar este ejemplo

 Copy Command

```
I = imread("peppers.png");  
imshow(I)  
title("Original Image")
```

 Get ▾

Original Image



Segmente la imagen en 50 regiones utilizando la agrupación K-medias. Devuelva la matriz de etiquetas L y las ubicaciones de los centroides del grupo C. Las ubicaciones de los centroides del grupo son los valores RGB de cada uno de los 50 colores.

```
[L,C] = imsegkmeans(I,50);
```

 Get ▼

Convierta la matriz de etiquetas en una imagen RGB. Especifique las ubicaciones de los centroides del grupo, C, como el mapa de colores para la imagen nueva.

```
J = label2rgb(L,im2double(C));
```

 Get ▼

Muestre la imagen cuantificada.

```
imshow(J)  
title("Color Quantized Image")
```

 Get ▼

Color Quantized Image



Escriba la imagen original y la comprimida en un archivo. El archivo de la imagen cuantificada tiene aproximadamente un cuarto del tamaño del archivo de la imagen original.

```
imwrite(I,"peppersOriginal.png");  
imwrite(J,"peppersQuantized.png");
```

 Get ▼


✓ Mejorar la segmentación K-medias utilizando un espacio de color alternativo

Lea y muestre una imagen de tejido teñido con hematoxilina y eosina (H&E). Este método de tinción ayuda a los patólogos a distinguir entre tipos de tejidos que están teñidos en azul-morado y rosa.

Probar este ejemplo

 [Copy Command](#)

```
he = imread("hestain.png");  
imshow(he)  
title("H&E Image");  
text(size(he,2),size(he,1)+15, ...  
      "Image courtesy of Alan Partin, Johns Hopkins University", ...  
      FontSize=7,HorizontalAlignment="right");
```

 Get ▼

H&E Image

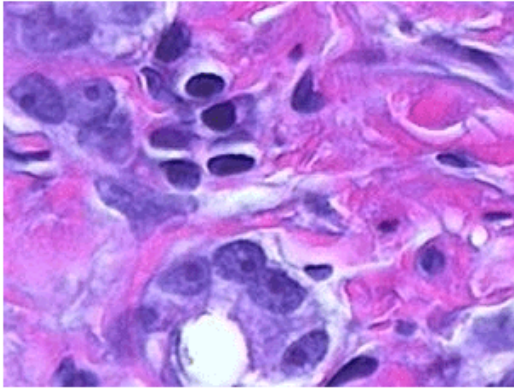


Image courtesy of Alan Partin, Johns Hopkins University

Convierta la imagen al espacio de color $L^*a^*b^*$ utilizando la función `rgb2lab`. El espacio de color $L^*a^*b^*$ separa la luminosidad y el color de la imagen. Esto hace que sea más fácil segmentar regiones por color, con independencia de la claridad.

```
lab_he = rgb2lab(he);
```

 Get ▼

Para segmentar la imagen utilizando solo información sobre el color, limite la imagen a los valores a^* y b^* en `lab_he`. Convierta la imagen al tipo de datos `single` para su uso con `imsegkmeans`. Utilice la función `imsegkmeans` para segmentar la imagen en tres regiones.

```
ab = lab_he(:,:,2:3);  
ab = im2single(ab);  
numColors = 3;  
L2 = imsegkmeans(ab,numColors);
```

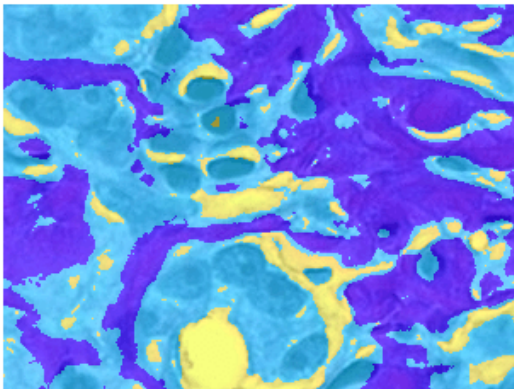
 Get ▼

Muestre la imagen de etiqueta como una superposición en la imagen original. La imagen de etiqueta separa las regiones de tejido teñidas de blanco, azul-morado y rosa.

```
B2 = labeloverlay(he,L2);  
imshow(B2)  
title("Labeled Image  $a^*b^*$ ")
```

 Get ▼

Labeled Image a^*b^*



Argumentos de entrada

contraer todo

I — Imagen que se desea segmentar



Imagen 2D en escala de grises | Imagen 2D en color | Imagen 2D multiespectral

Imagen que se desea segmentar, especificada como imagen 2D en escala de grises, imagen 2D en color o imagen 2D multiespectral. Si la imagen original es de tipo de datos `double`, convierta la imagen al tipo de datos `single` utilizando la función `im2single`.

Tipos de datos: `single` | `int8` | `int16` | `uint8` | `uint16`

✓ **k – Número de grupos**
entero positivo

Número de grupos que se desea crear, especificado como entero positivo.

Argumentos de par nombre-valor

Especifique pares de argumentos opcionales como `Name1=Value1, ..., NameN=ValueN`, donde `Name` es el nombre del argumento y `Value` es el valor correspondiente. Los argumentos de nombre-valor deben aparecer después de otros argumentos. Sin embargo, el orden de los pares no importa.

Ejemplo: `imsegkmeans(I,k,NumAttempts=5)` repite el proceso de agrupación cinco veces.

En las versiones anteriores a la R2021a, utilice comas para separar cada nombre y valor, y encierre `Name` entre comillas.

Ejemplo: `imsegkmeans(I,k,"NumAttempts",5)` repite el proceso de agrupación cinco veces.

✓ **NormalizeInput – Normalizar datos de entrada**
`true` o `1` (predeterminado) | `false` o `0`

Normalizar datos de entrada a una media cero y una varianza unitaria, especificado como `1` (`true`) o `0` (`false`) numérico o lógico. Si especifica `true`, `imsegkmeans` normaliza cada canal de la entrada individualmente.

✓ **NumAttempts – Número de veces que se repetirá el proceso de agrupación**
`3` (predeterminado) | entero positivo

Número de veces que se repetirá el proceso de agrupación utilizando nuevas ubicaciones iniciales de los centroides de los grupos, especificado como entero positivo.

✓ **MaxIterations – Número máximo de iteraciones**
`100` (predeterminado) | entero positivo

Número máximo de iteraciones, especificado como entero positivo.

✓ **Threshold – Umbral de precisión**
`1e-4` (predeterminado) | número positivo

Umbral de precisión, especificado como número positivo. El algoritmo se detiene cuando cada uno de los centros de grupo se mueve menos que el valor del umbral en iteraciones consecutivas.

▼ L — Matriz de etiquetas

Matriz de enteros positivos

Matriz de etiquetas, especificada como matriz de enteros positivos. Los píxeles con la etiqueta 1 pertenecen al primer grupo, con la etiqueta 2, al segundo grupo, y así sucesivamente para cada uno de los k grupos. L tiene las mismas dos primeras dimensiones que la imagen I . El tipo de datos de L depende del número de grupos.

Tipo de datos de L	Número de grupos
uint8	$k \leq 255$
uint16	$256 \leq k \leq 65535$
uint32	$65536 \leq k \leq 2^{32}-1$
double	$2^{32} \leq k$

▼ centers — Ubicaciones de los centroides de los grupos

Matriz numérica

Ubicaciones de los centroides de los grupos, devueltas como matriz numérica de tamaño k por c , donde k es el número de grupos y c es el número de canales. `centers` es del mismo tipo de datos que la imagen I .

Sugerencias

- La función produce resultados reproducibles. La salida no varía en múltiples ejecuciones dado que los argumentos de entrada son los mismos.
- La función `imsegkmeans` acepta imágenes de entrada en todos los espacios de color compatibles. Utilizar un espacio de color diferente genera resultados diferentes. Si no recibe resultados satisfactorios para una imagen de entrada, considere intentarlo con un espacio de color alternativo. Para obtener más información sobre los espacios de color en MATLAB®, consulte [Comprender los espacios de color y la conversión entre ellos](#).
- Para realizar una agrupación K-medias con las imágenes de tipo de datos `double`, convierta la imagen al tipo de datos `single` utilizando la función `im2single`. En las aplicaciones que requieren datos de tipo `double`, consulte la función `kmeans` (Statistics and Machine Learning Toolbox).

Referencias

[1] Arthur, David, and Sergei Vassilvitskii. "K-Means++: The Advantages of Careful Seeding." In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, 1027–35. SODA '07. USA: Society for Industrial and Applied Mathematics, 2007.

Historial de versiones

Introducido en R2018b

Consulte también

Apps

Image Segmenter

Funciones

`imsegkmeans3` | `gabor` | `imgaborfilt` | `labeloverlay` | `label2rgb` | `superpixels` | `lazysnapping` | `watershed` | `labelmatrix`

Temas

Segmentación basada en colores utilizando la agrupación K-medias