

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.

Segmentación basada en colores utilizando la agrupación K-medias

Este ejemplo muestra cómo segmentar colores de forma automatizada utilizando la agrupación K-medias.

La agrupación es una forma de separar grupos de objetos. La agrupación K-medias trata cada objeto como si tuviera una ubicación en el espacio. Encuentra particiones tales que los objetos de cada grupo están tan próximos unos a otros como sea posible y tan lejos de objetos de otros grupos como sea posible. Puede utilizar la función `imsegkmeans` para separar píxeles de imagen por valor en grupos en un espacio de color. Este ejemplo realiza un agrupamiento K-medias de una imagen en los espacios de color RGB y $L^*a^*b^*$ para mostrar cómo utilizar diferentes espacios de color puede mejorar los resultados de segmentación.

Probar este ejemplo

 Copy Command

Paso 1: Leer una imagen

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.

```
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

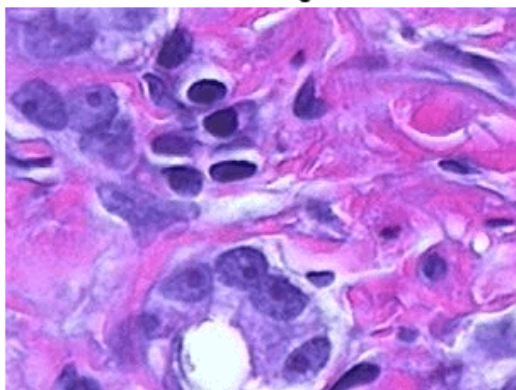


Image courtesy of Alan Partin, Johns Hopkins University

Paso 2: Clasificar colores en el espacio de color RGB utilizando la agrupación K-medias

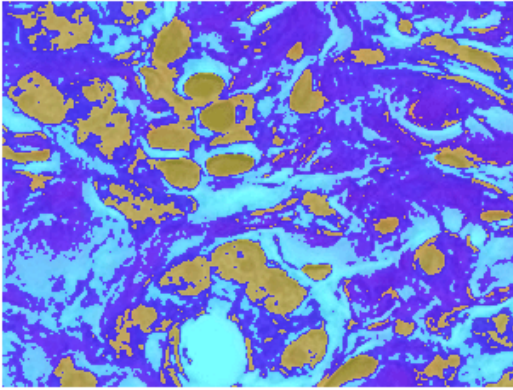
Segmente la imagen en tres regiones utilizando la agrupación K-medias en el espacio de color RGB. Para cada píxel de la imagen de entrada, la función `imsegkmeans` devuelve una etiqueta correspondiente a un grupo.

Muestre la imagen de etiqueta como una superposición en la imagen original. La imagen de etiqueta agrupa de forma errónea las regiones blanca, azul claro-morada y rosa claro. Puesto que el espacio de color RGB combina la información de brillo y color en cada canal (rojo, verde, azul), las versiones más claras de dos colores diferentes están más cerca entre ellas y resultan más difíciles de segmentar que las versiones más oscuras de los mismos dos colores.

```
numColors = 3;  
L = imsegkmeans(he,numColors);  
B = labeloverlay(he,L);  
imshow(B)  
title("Labeled Image RGB")
```

 Get ▾

Labeled Image RGB



Paso 3: Convertir una imagen del espacio de color RGB al espacio de color L*a*b*

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. El espacio de color también es más consistente con la percepción visual humana de las regiones blanca, azul-morada y rosa distintas de la imagen.

El espacio de color L*a*b* se deriva de los valores triestímulos CIE XYZ. El espacio L*a*b* consta de la capa de luminosidad L*, de la capa de cromaticidad a* que indica dónde se sitúa un color en el eje rojo-verde y de la capa de cromaticidad b* que indica dónde se sitúa un color en el eje azul-amarillo. Toda la información sobre el color está en las capas a* y b*.

Convierta la imagen al espacio de color L*a*b* utilizando la función `rgb2lab`.

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

 Get ▼

Paso 4: Clasificar colores en un espacio a*b* utilizando la agrupación K-medias

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 la función `imsegkmeans`. Utilice la función `imsegkmeans` para separar los píxeles de imagen en tres grupos. Establezca el valor del argumento nombre-valor `NumAttempts` de forma que repita la agrupación tres veces con diferentes posiciones de los centroides del grupo inicial para evitar el ajuste a un mínimo local.

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

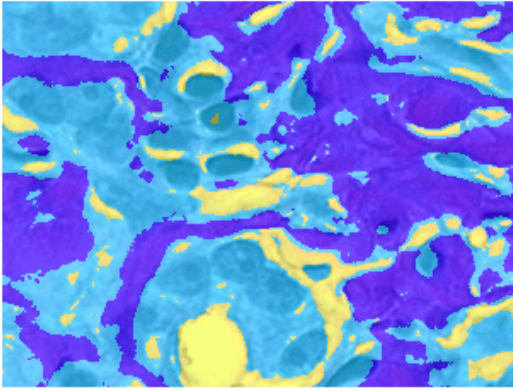
 Get ▼

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

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

 Get ▼

Labeled Image a*b*



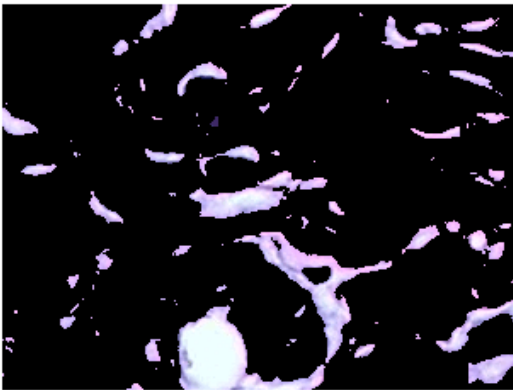
Paso 5: Crear imágenes que segmenten una imagen HyE por color

Utilizando `pixel_labels`, puede separar objetos en la imagen original `hestain.png` por color, lo que resulta en tres imágenes enmascaradas.

```
mask1 = pixel_labels == 1;
cluster1 = he.*uint8(mask1);
imshow(cluster1)
title("Objects in Cluster 1");
```

 Get ▼

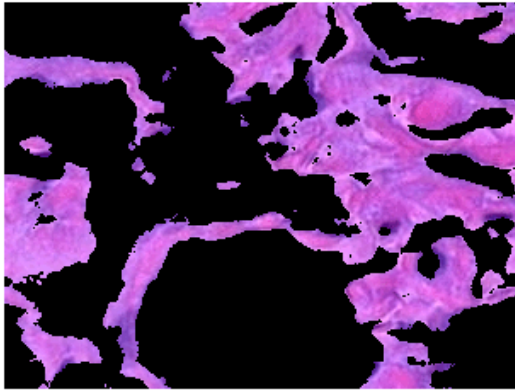
Objects in Cluster 1



```
mask2 = pixel_labels == 2;
cluster2 = he.*uint8(mask2);
imshow(cluster2)
title("Objects in Cluster 2");
```

 Get ▼

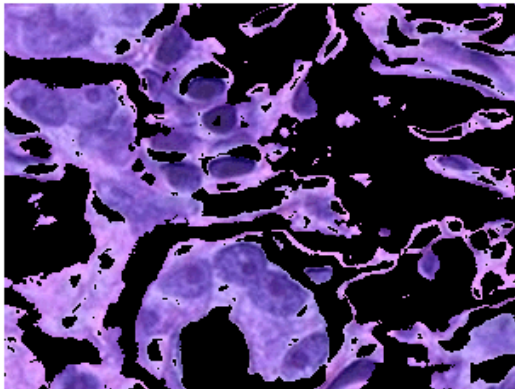
Objects in Cluster 2



```
mask3 = pixel_labels == 3;
cluster3 = he.*uint8(mask3);
imshow(cluster3)
title("Objects in Cluster 3");
```

Get ▼

Objects in Cluster 3



Paso 6: Segmentar núcleos

El grupo 3 solo contiene los objetos azules. Observe que hay objetos azul oscuro y azul claro. Puede separar el azul oscuro del azul claro utilizando la capa L^* en el espacio de color $L^*a^*b^*$. Los núcleos de células son azul oscuro.

La capa L^* contiene el valor de brillo de cada píxel. Extraiga los valores de brillo de los píxeles de este grupo y determine su umbral con un umbral global utilizando la función `imbinarize`. La máscara `idx_light_blue` proporciona los índices de píxeles azul claro.

```
L = lab_he(:,:,1);
L_blue = L.*double(mask3);
L_blue = rescale(L_blue);
idx_light_blue = imbinarize(nonzeros(L_blue));
```

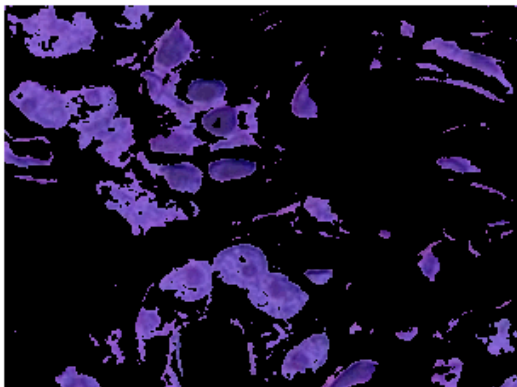
Get ▼

Copie la máscara de objetos azules, `mask3`, y elimine los píxeles azul claro de la máscara. Aplique la máscara nueva a la imagen original y muestre el resultado. Solo son visibles los núcleos de células azul oscuro.

```
blue_idx = find(mask3);  
mask_dark_blue = mask3;  
mask_dark_blue(blue_idx(idx_light_blue)) = 0;  
  
blue_nuclei = he.*uint8(mask_dark_blue);  
imshow(blue_nuclei)  
title("Blue Nuclei")
```

 Get ▼

Blue Nuclei



Consulte también

[imsegkmeans](#)

Temas relacionados

- Comprender los espacios de color y la conversión entre ellos
- Segmentación por colores utilizando el espacio de color L*a*b