الجلسة الحادية عشر

Segmentation & Clustering

1.1 Clustering-based Segmentation (K-Means)

تعتبر خوارزمية K-Means أحد أشهر خوارزميات العنقدة Clustering والتي يمكن استخدامها في عمليات التجزيء اللوني للصور حيث نفترض عدد العناقيد المراد تنفيذ الخوارزمية عليها مساوي لعدد الأصناف اللونية المراد الحصول عليها في الصورة.

مشكلة هذه الخوارزمية أنها تحتاج تحديد مراكز عناقيد ابتدائية أو ضبطها بشكل عشوائي كما أنها تحتاج لتحديد عدد الأصناف أو العناقيد.

والآن ننفذ الكود التالي:

```
import cv2
from sklearn.cluster import KMeans
import matplotlib.pyplot as plt
import numpy as np

# Load the image
image = cv2.imread('3.jpg')
image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)

# Reshape the image to be a list of RGB values
pixels = image.reshape(-1, 3)

# Perform K-Means
kmeans = KMeans(n_clusters=5)  # Change the number of clusters as needed
kmeans.fit(pixels)

# Replace each pixel with its centroid
```

```
segmented_img = kmeans.cluster_centers_[kmeans.labels_]
segmented_img = segmented_img.reshape(image.shape)

# Convert data types for imshow
image = image.astype(np.uint8)
segmented_img = segmented_img.astype(np.uint8)

# Create a subplot
fig, ax = plt.subplots(1, 2, figsize=(10, 5))

# Show original image
ax[0].imshow(image)
ax[0].set_title('Original Image')

# Show segmented image
ax[1].imshow(segmented_img)
ax[1].set_title('Segmented Image')

plt.show()
```

1.2 Clustering-based Segmentation (Mean Shift)

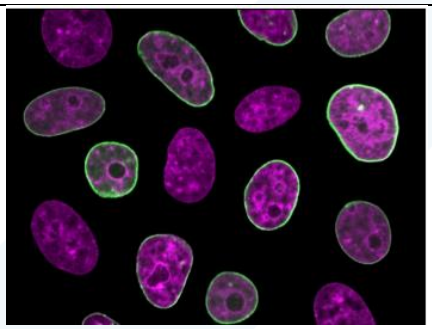
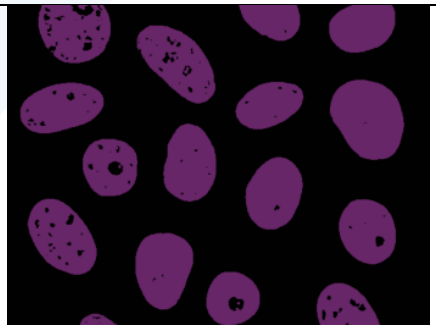
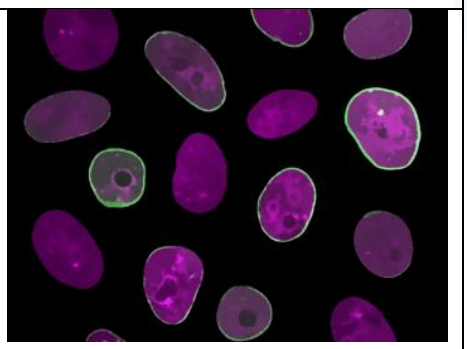
هي خوارزمية تجزيء أخرى تعتمد مبدأ تحريك المتوسط. سهولتها أنها لا تحتاج لتحديد مراكز عناقيد ابتدائية.

نفذ الكود:

```
import numpy as np
import cv2
import matplotlib.pyplot as plt

# Load the image
image = cv2.imread('3.jpg')
```

```
image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)


# Apply Mean Shift

mean_shifted = cv2.pyrMeanShiftFiltering(image, sp=15, sr=50)


# Create a subplot

fig, ax = plt.subplots(1, 2, figsize=(10, 5))


# Show original image

ax[0].imshow(image)

ax[0].set_title('Original Image')


# Show segmented image

ax[1].imshow(mean_shifted)

ax[1].set_title('Segmented Image')


plt.show()
```

نتائج التنفيذ:

Original Image	K-Means	Mean Shift
		

1.3 خوارزمية Marker Watershed

هي خوارزمية تجزيء Top-down

خطواتها:

1- **تحديد العلامات Markers:** العلامات هي بعض النقاط المحددة داخل الصورة التي تمثل الأماكن التي نعرف بالتأكيد أنها تنتمي إلى الخلفية أو الأجسام.

2- **تحويل المسافة Distance Transform:** يتم حساب تحويل المسافة للصورة. تحويل المسافة هو صورة تمثل كمية البكسلات التي يجب الانتقال عبرها للوصول إلى البكسل الأقرب من القيمة الأقل.

3- **تطبيق خوارزمية Watershed:** بمجرد تحديد العلامات وحساب تحويل المسافة، يمكن تطبيق خوارزمية Watershed.

```
import cv2

import numpy as np

import matplotlib.pyplot as plt


# Load the image in grayscale for thresholding and in color for watershed

image_gray = cv2.imread('1.jpg', 0)

image_color = cv2.imread('1.jpg')

image_color = cv2.cvtColor(image_color, cv2.COLOR_BGR2RGB)
```

```
# Apply adaptive thresholding
_, thresh = cv2.threshold(image_gray, 0, 255, cv2.THRESH_BINARY_INV + cv2.THRESH_OTSU)

# Noise removal
kernel = np.ones((3,3),np.uint8)
opening = cv2.morphologyEx(thresh, cv2.MORPH_OPEN, kernel, iterations = 2)

# Sure background area
sure_bg = cv2.dilate(opening, kernel, iterations=3)

# Finding sure foreground area
dist_transform = cv2.distanceTransform(opening, cv2.DIST_L2, 5)
_, sure_fg = cv2.threshold(dist_transform, 0.9*dist_transform.max(), 255, 0)

# Finding unknown region
sure_fg = np.uint8(sure_fg)
unknown = cv2.subtract(sure_bg, sure_fg)

# Marker labelling
_, markers = cv2.connectedComponents(sure_fg)

# Add one to all labels so that sure background is not 0, but 1
markers = markers+1

# Now, mark the region of unknown with zero
markers[unknown==255] = 0
```

```
# Apply watershed to the color image
markers = cv2.watershed(image_color, markers)
image_color[markers == -1] = [255,0,0]

# Create a subplot
fig, ax = plt.subplots(1, 2, figsize=(10, 5))

# Show original image
ax[0].imshow(cv2.cvtColor(cv2.imread('1.jpg'), cv2.COLOR_BGR2RGB))
ax[0].set_title('Original Image')

# Show segmented image
ax[1].imshow(markers, cmap='jet')  # The segmented image
ax[1].set_title('Segmented Image')

plt.show()
```

نتيجة التنفيذ: