

به نام خدا

گزارش کار پروژه درس یادگیری ماشین

...: عنوان گزارش کار ...:

قطعه بندی تصاویر با استفاده از الگوریتم K-Means

استاد محترم : جناب دکتر آهنگران

تهیه کننده : امین زایراومالی

دانشجو کارشناسی ارشد مهندسی نرم افزار

دانشگاه علوم و فنون مازندارن

تعریف پروژه : ۳ تصویر مشخص داده شده را بایستی با استفاده از الگوریتم K-Means برای تعداد خوشه های ۲ و ۳ و ۵ عددی خوشه بندی نموده و تصاویر خروجی خوشه بندی شده را نمایش داده و سپس میزان فواصل درون خوشه ای و برون خوشه ای تصاویر را با استفاده از نمودار نمایش دهید.

مقدمه : از مهمترین تکنیک های عملی داده کاوی که کاربرد زیادی در علوم مختلف دارد، می توان به خوشه بندی کی مین اشاره کرد خوشه بندی کی مین یک الگوریتم یادگیری ماشینی بدون نظارت است که هدف آن تقسیم تعداد مشخص مشاهدات به خوشه های (K) است که در آن هر مشاهده متعلق به خوشه ای با نزدیک ترین میانگین است. خوشه به مجموعه ای از نقاط داده اشاره دارد که به دلیل شباهت های خاص با هم جمع شده اند. برای تقسیم بندی تصویر، خوشه ها در اینجا رنگ های تصویر متفاوتی دارند.

پیاده سازی پروژه با زبان پایتون انجام شد به شرح کدهای زیر می باشد :

۱- مرحله اول و دوم : اضافه کردن کتابخانه های مورد نیاز و بارگزاری تصاویر

مرحله اول : اضافه کردن کتابخانه های مورد نیاز

```
In [25]: import cv2 as cv          # کتابخانه opencv
import numpy as np              # کتابخانه numpy
import matplotlib.pyplot as plt  # کتابخانه matplotlib
import matplotlib.image as mpimg # کتابخانه Ipython
```

مرحله دوم : لود کردن اطلاعات تصاویر در متغیر های مربوط به خودشان

```
In [26]: # خواندن سه تصویر و قرار دادن در متغیر مربوطه اش
Original_img1 = mpimg.imread('image-1.jpg')
Original_img2 = mpimg.imread('image-2.jpg')
Original_img3 = mpimg.imread('image-3.jpg')
```

۲- مرحله سوم : نمایش تصاویر اصلی بارگزاری شده

مرحله سوم : نمایش تصاویر اصلی

```
In [27]: # نمایش ۳ تصویر داده شده
fig = plt.figure(figsize=(10, 7))
rows = 2
columns = 2
fig.add_subplot(rows, columns, 1)
plt.imshow(Original_img1)
plt.axis('off')
plt.title("First")
fig.add_subplot(rows, columns, 2)
plt.imshow(Original_img2)
plt.axis('off')
plt.title("Second")
fig.add_subplot(rows, columns, 3)
plt.imshow(Original_img3)
plt.axis('off')
plt.title("Third")
```

Out[27]: Text(0.5, 1.0, 'Third')

First



Second



Third



۳- مرحله چهارم تا هفتم : اطلاعات تصاویر بارگزاری شده را به صورت آرایه دو بعدی با رنگ بندی RGB تبدیل نموده و سپس تبدیل به نوع اعشاری می‌کنیم و بعد تصاویر ۳ رنگی را نمایش می‌دهیم

RGB - 2 Dimention مرحله چهارم : تبدیل تصاویر به صورت دو بعدی در فرمت سه رنگ آر جی بی

```
In [4]: # RGB - 2D تبدیل تصاویر به دو بعدی و ۳ رنگ اصلی آر جی بی
image1 = Original_img1.reshape((-1,3))
image2 = Original_img2.reshape((-1,3))
image3 = Original_img3.reshape((-1,3))
```

Float مرحله ششم : تبدیل اطلاعات تصاویر به نوع فلویت

```
In [5]: image1 = np.float32(image1)
image2 = np.float32(image2)
image3 = np.float32(image3)
```

مرحله هفتم : نمایش میزان پیکسل ردیف اطلاعات تصاویر

```
In [6]: # نمایش میزان پیکسل ردیف اطلاعات تصاویر
print("Image 1 : ", image1.shape)
print("Image 2 : ", image2.shape)
print("Image 3 : ", image3.shape)

Image 1 : (77440, 3)
Image 2 : (150528, 3)
Image 3 : (231852, 3)
```

مرحله هشتم : تعیین تعداد خوشه بندی تصاویر برای سه میزان کا به تعداد ۲ و ۳ و ۵

```
In [7]: k1 = 2
k2 = 3
k3 = 5
```

۴ - مرحله هشتم تا دهم : برای سه مقدار K تعیین شده الگوریتم را بروی تصاویر اجرا می‌کنیم و اطلاعات خروجی شامل نقاط مرکز و لیبل ها و مقادیر درون خوشه ای و برون خوشه ای و مقدار تابع هدف مینیمم را دریافت می‌کنیم

```
jupyter ML-Project Last Checkpoint: 20 minutes ago (autosaved)

File Edit View Insert Cell Kernel Widgets Help

In [7]: k1 = 2
k2 = 3
k3 = 5

مرحله هشتم : تعیین تعداد خوشه بندی تصاویر برای سه میزان کا به تعداد ۲ و ۳ و ۵

In [8]: # این معیار جانم تکرار است. وقتی این معیار برآورده شد، تکرار الگوریتم متوقف می‌شود.
# علت استفاده برای این است که در حلقه متنیماال بی یا بان قرار نگیرد.
criteria = (cv.TERM_CRITERIA_EPS + cv.TERM_CRITERIA_MAX_ITER, 10, 1.0)

# خوشه بندی تصویر اول با سه مقدار کا تعیین شده
ret1_k1,label1_k1,center1_k1=cv.kmeans(image1,k1,None,criteria,10,cv.KMEANS_RANDOM_CENTERS)
ret1_k2,label1_k2,center1_k2=cv.kmeans(image2,k2,None,criteria,10,cv.KMEANS_RANDOM_CENTERS)
ret1_k3,label1_k3,center1_k3=cv.kmeans(image3,k3,None,criteria,10,cv.KMEANS_RANDOM_CENTERS)

# خوشه بندی تصویر دوم با سه مقدار کا تعیین شده
ret2_k1,label2_k1,center2_k1=cv.kmeans(image2,k1,None,criteria,10,cv.KMEANS_RANDOM_CENTERS)
ret2_k2,label2_k2,center2_k2=cv.kmeans(image2,k2,None,criteria,10,cv.KMEANS_RANDOM_CENTERS)
ret2_k3,label2_k3,center2_k3=cv.kmeans(image2,k3,None,criteria,10,cv.KMEANS_RANDOM_CENTERS)

# خوشه بندی تصویر سوم با سه مقدار کا تعیین شده
ret3_k1,label3_k1,center3_k1=cv.kmeans(image3,k1,None,criteria,10,cv.KMEANS_RANDOM_CENTERS)
ret3_k2,label3_k2,center3_k2=cv.kmeans(image3,k2,None,criteria,10,cv.KMEANS_RANDOM_CENTERS)
ret3_k3,label3_k3,center3_k3=cv.kmeans(image3,k3,None,criteria,10,cv.KMEANS_RANDOM_CENTERS)

مرحله دهم : تبدیل داده های خوشه بندی به تصاویر

In [9]: # Now convert back into uint8, and make original image
# تبدیل تصویر اول به ارایری سه مقدار کا
center1_k1 = np.uint8(center1_k1)
center1_k2 = np.uint8(center1_k2)
center1_k3 = np.uint8(center1_k3)

# تبدیل تصویر دوم به ارایری سه مقدار کا
center2_k1 = np.uint8(center2_k1)
center2_k2 = np.uint8(center2_k2)
center2_k3 = np.uint8(center2_k3)
```

۴ - مرحله دهم تا دوازدهم : براساس اطلاعات خروجی الگوریتم تصاویر خوشه بندی شده با میزان K های خواسته شده مجدد بازسازی و ذخیره می نمایید تا بتوانیم نمایش دهیم

مرحله دهم : تبدیل داده های خوشه بندی به تصاویر

```
In [9]: # Now convert back into uint8, and make original image   ابعاد عکسهای اورجینال از داده های خوشه بندی شده

# تبدیل تصویر اول به ازاری سه مقدار کا
center1_k1 = np.uint8(center1_k1)
center1_k2 = np.uint8(center1_k2)
center1_k3 = np.uint8(center1_k3)

# تبدیل تصویر دوم به ازاری سه مقدار کا
center2_k1 = np.uint8(center2_k1)
center2_k2 = np.uint8(center2_k2)
center2_k3 = np.uint8(center2_k3)

# تبدیل تصویر سوم به ازاری سه مقدار کا
center3_k1 = np.uint8(center3_k1)
center3_k2 = np.uint8(center3_k2)
center3_k3 = np.uint8(center3_k3)
```

مرحله یازدهم : تبدیل ابعاد و ذخیره های تصاویر

```
In [10]: # convert all pixels to the color of the centroids   تبدیل مرکز های خوشه بندی به رنگ بندی تصاویر

img1_k1 = center1_k1[label1_k1.flatten()]
img1_k2 = center1_k2[label1_k2.flatten()]
img1_k3 = center1_k3[label1_k3.flatten()]

img2_k1 = center2_k1[label2_k1.flatten()]
img2_k2 = center2_k2[label2_k2.flatten()]
img2_k3 = center2_k3[label2_k3.flatten()]

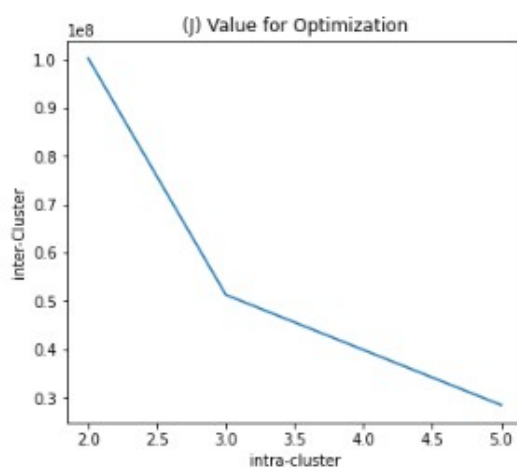
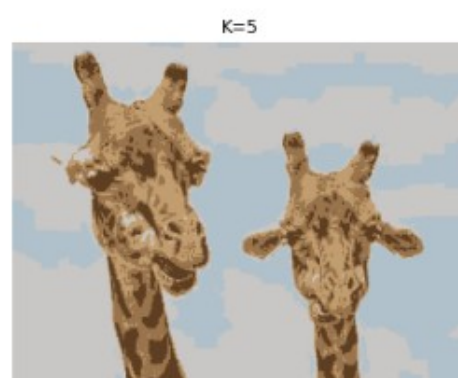
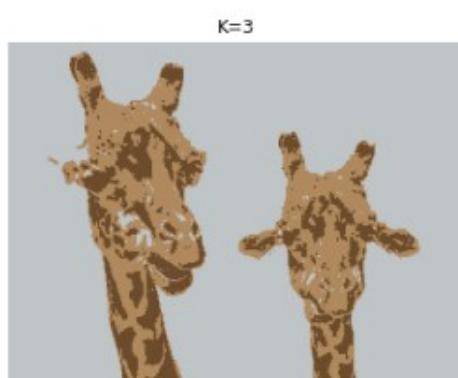
img3_k1 = center3_k1[label3_k1.flatten()]
img3_k2 = center3_k2[label3_k2.flatten()]
img3_k3 = center3_k3[label3_k3.flatten()]

# reshape back to the original image dimension   برگرداندن ابعاد تصاویر به ابعاد اصلی
img1_k1_2 = img1_k1.reshape((Original_img1.shape))
img1_k2_2 = img1_k2.reshape((Original_img1.shape))
img1_k3_2 = img1_k3.reshape((Original_img1.shape))

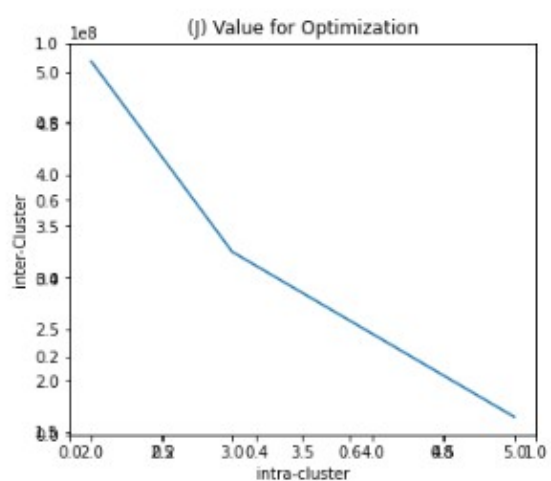
img2_k1_2 = img2_k1.reshape((Original_img2.shape))
img2_k2_2 = img2_k2.reshape((Original_img2.shape))
img2_k3_2 = img2_k3.reshape((Original_img2.shape))

img3_k1_2 = img3_k1.reshape((Original_img3.shape))
img3_k2_2 = img3_k2.reshape((Original_img3.shape))
img3_k3_2 = img3_k3.reshape((Original_img3.shape))
```

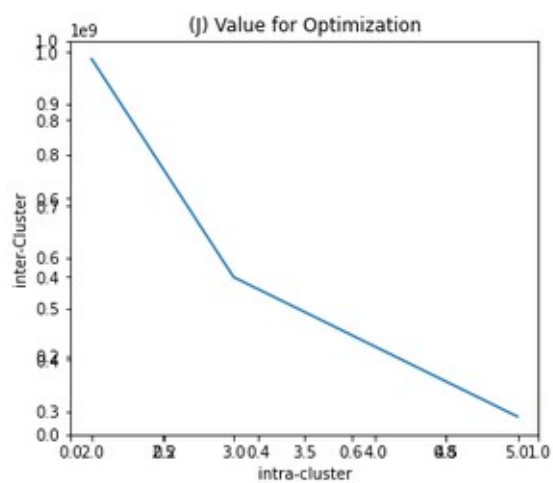
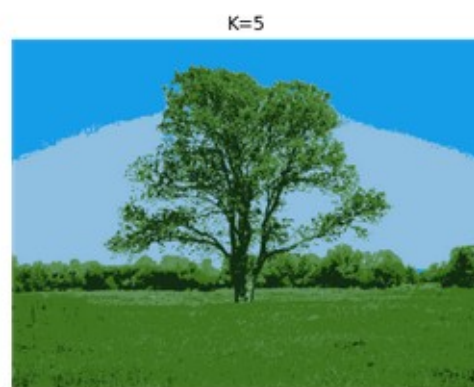
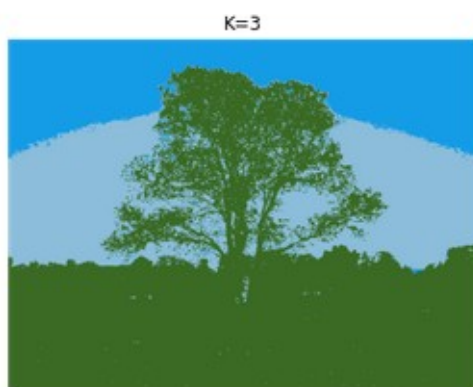
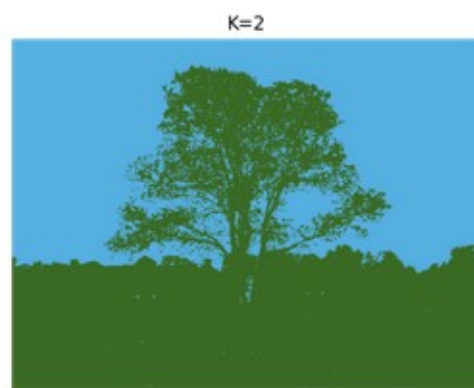
۵ - مراحل بعدی تا پایان : نتایج تصاویر خوشه بندی شده را به همراه نمودار اطلاعات درون خوشه ای و برون خوشه ای براساس مقادیر تابع مینیم نمایش می دهیم که به شرح تصاویر زیر می باشند



نمایش نتایج مربوط به تصویر اول به ازای K مختلف



نمایش نتایج مربوط به تصویر دوم به ازای K مختلف



نمایش نتایج مربوط به تصویر سوم به ازای K مختلف

جمع بندی نتایج پروژه

تصاویر را در متغیرهایی بارگذاری کردیم و سپس آن را به آرایه دو بعدی با رنگ بندی آر جی بی تغییر دادیم

سپس با استفاده از الگوریتم کی مین برای سه مقدار ۲ و ۳ و ۵ خوشه بندی نمودیم

سپس داده هایی که خوشه بندی کردیم را به تصاویر قابل مشاهده بازسازی یا تبدیل نمودیم به ازای کای مشخص آن نشان دادیم همچنین در پایان نیز میزان پارامترهای درون خوشه ای و برون خوشه ای را طبق خوشه بندی الگوریتم کی مین بروی نمودار نشان دادیم

با تشکر فراوان از استاد محترم : جناب دکتر آهنگران

تهیه کننده : امین زایراومالی

دانشجو کارشناسی ارشد مهندسی کامپیوتر نرم افزار

دانشگاه علوم و فنون مازنداران