

Mã sinh viên : 20020688

Họ và tên : Đỗ Đức Mạnh

Bài thực hành số 7

Môn : Xử lý ảnh và thị giác robot

Đề bài: Cho bộ dữ liệu các loại bướm.

1. Sử dụng Kmeans Clustering để quantization màu sắc (8,16,64,256)
2. Trích xuất đặc trưng, sau đó dùng KNN hoặc SVM để phân loại.
3. So sánh kết quả sử dụng H-histogram của HSV

Bài làm

B1: Đọc đường dẫn đến bộ dữ liệu

Sử dụng thư viện 'os' để truy cập vào đường dẫn bộ dữ liệu.

```
In [2]: data_path = './leedsbutterfly_dataset_v1/leedsbutterfly'
        class_name = ['images']

In [3]: def get_list_files(dirName):
        files_list = os.listdir(dirName)
        return files_list

In [4]: file_lists = get_list_files(data_path + '/' + class_name[0])

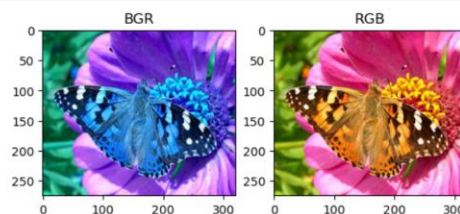
In [5]: len(file_lists)
Out[5]: 832
```

Hiển thị một hình ảnh ngẫu nhiên trong bộ dữ liệu:

```
In [6]: def rand_img(file_lists_img):
        rand_img = np.random.randint(0, len(file_lists_img))
        img_dir = data_path + '/' + class_name[0] + '/' + file_lists[rand_img]
        return img_dir

In [7]: rand_img(file_lists)
Out[7]: './leedsbutterfly_dataset_v1/leedsbutterfly/images/0090011.png'

In [8]: fig, (ax1, ax2) = plt.subplots(1, 2)
        rand_img_dir = rand_img(file_lists)
        imgBGR = cv2.imread(rand_img_dir)
        imgRGB = cv2.cvtColor(imgBGR, cv2.COLOR_BGR2RGB)
        ax1.imshow(imgBGR)
        ax2.imshow(imgRGB)
        ax1.title.set_text('BGR')
        ax2.title.set_text('RGB')
        plt.show()
```



B2: Lựa chọn ra một số ảnh để lấy mẫu màu sắc.

Bộ dữ liệu có 10 loại bướm, mỗi loại chọn ra 1 bức ảnh để lấy mẫu.

Đưa bức ảnh về ma trận 1 chiều và giữ nguyên không gian màu.

```
In [9]: img_sample = ['0010004.png', '0020003.png', '0030040.png', '0040002.png', '0050008.png', '0060054.png', '0070034.png', '0080057.png', '0090001.png', '0100000.png']

In [10]: imgs_arr=[]
for i in img_sample:
    img = cv2.imread(data_path+'/'+class_name[0]+'/'+'i')
    imgRGB= cv2.cvtColor(img,cv2.COLOR_BGR2RGB)
    imgRGB=np.array(imgRGB,dtype=np.float64)/255
    # print(imgRGB.shape)
    w,h,d=tuple(imgRGB.shape)
    img_arr = np.reshape(imgRGB,(w* h, d))
    imgs_arr.append(img_arr)
    print(img_arr.shape)
# plt.imshow(img)
# plt.show()

(504000, 3)
(255036, 3)
(208000, 3)
(90560, 3)
(227000, 3)
(86400, 3)
(260928, 3)
(14850, 3)
(67500, 3)
(82500, 3)
```

B3: Lấy mẫu màu sắc với 10 ảnh mẫu.

Tổng hợp 10 ảnh mẫu vào một ma trận:

```
In [11]: All_pixels = np.concatenate((imgs_arr[0],imgs_arr[1],imgs_arr[2],imgs_arr[3],imgs_arr[4],imgs_arr[5],
                                     imgs_arr[6],imgs_arr[7],imgs_arr[8],imgs_arr[9]))
```

```
In [31]: All_pixels.shape
```

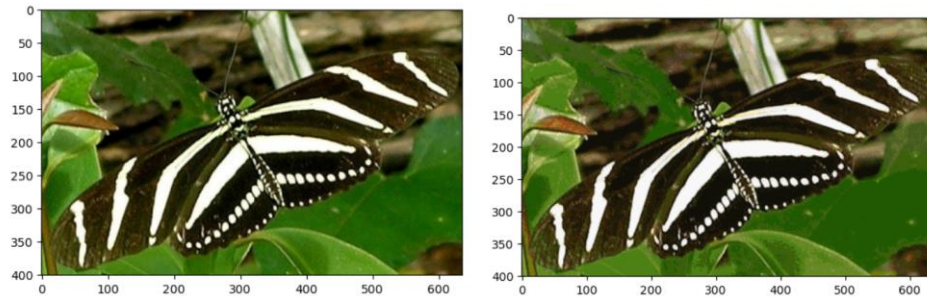
```
Out[31]: (1796774, 3)
```

Sử dụng Kmeans Clustering để phân đoạn màu sắc thành 64 màu, sử dụng 1500 pixel trong ma trận tổng hợp.

```
In [94]: image_array_sample = shuffle(All_pixels, random_state=0, n_samples=200000)
num_colors = 64
kmeans = KMeans(n_clusters=num_colors, n_init="auto", random_state=0).fit(
    image_array_sample)
```

```
In [95]: labels1 = kmeans.predict(imgs_arr[1])
```

Hình ảnh trước và sau khi phân đoạn màu sắc:



Trước

Sau

Color Histogram của ảnh sau khi phân đoạn màu sắc:

```
: color_histogram1
: array([0.09770777, 0.0040112 , 0.00295252, 0.03000361, 0.00130962,
        0.00721075, 0.05255337, 0.00254082, 0.          , 0.00058423,
        0.00801063, 0.02715303, 0.00929281, 0.00320347, 0.          ,
        0.00183504, 0.0108965 , 0.02165577, 0.01671529, 0.0660691 ,
        0.002631 , 0.00035681, 0.          , 0.01730736, 0.12110447,
        0.0003176 , 0.          , 0.01881303, 0.00202324, 0.1146348 ,
        0.00390533, 0.00443859, 0.00155664, 0.00657162, 0.02211845,
        0.00939867, 0.00163114, 0.00066265, 0.00659123, 0.00158017,
        0.          , 0.0008587 , 0.05167898, 0.07378566, 0.          ,
        0.06343026, 0.00252121, 0.00403865, 0.          , 0.00124296,
        0.01737402, 0.00792829, 0.01048087, 0.          , 0.00242711,
        0.01403723, 0.00610502, 0.00185072, 0.01290406, 0.02295362,
        0.00041563, 0.00321915, 0.00339952])
```

B4: Gán nhãn cho tất cả các ảnh trong tập dữ liệu để đưa vào mô hình học máy.

Đưa vào học máy sẽ là các ma trận color histogram của ảnh sau khi phân đoạn và nhãn của 10 loại bướm(1→10).

```
In [131]: target=[]
          images=[]
          flat_data1=[]
          flat_data2=[]

          path = os.path.join(data_path,class_name[0])
          for img in os.listdir(path):
              imgBGR = cv2.imread(os.path.join(path,img))
              imgRGB = cv2.cvtColor(imgBGR,cv2.COLOR_BGR2RGB)
              imgRGB = np.array(imgRGB,dtype=np.float64)/255
              w, h, d = tuple(imgRGB.shape)
              img_arr = np.reshape(imgRGB,(w*h, d))
              clus_img = kmeans.predict(img_arr)
              new_img = recreate_img(kmeans.cluster_centers_, clus_img, w, h)
              color_histogram = np.histogram(clus_img, bins=np.arange(0,8))[0]/(w*h)
              # new_img=np.array(new_img,dtype=np.float64)/255

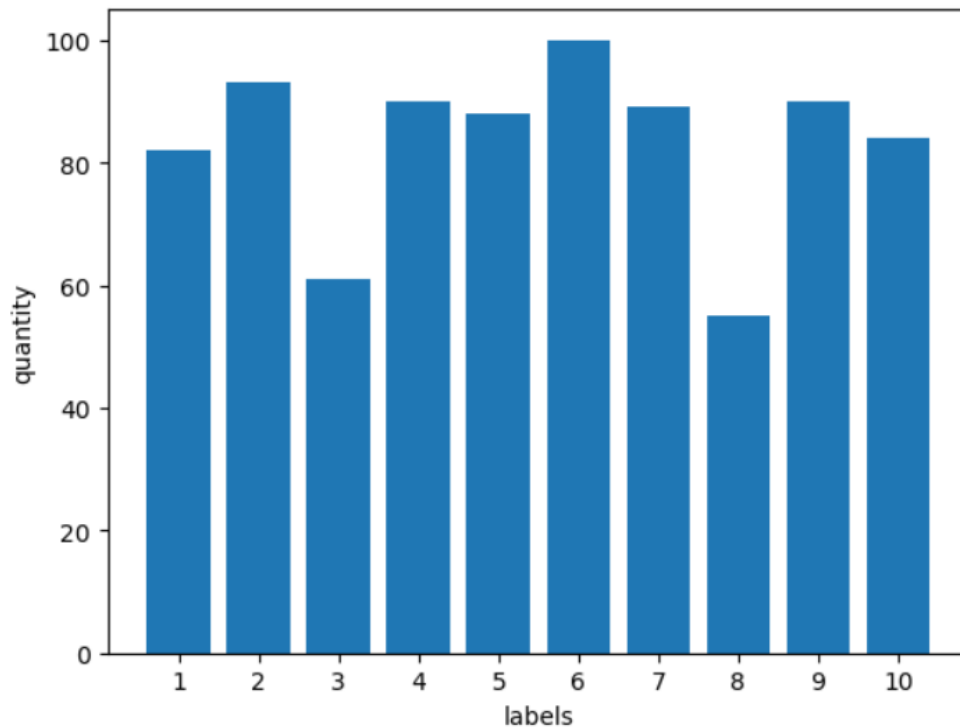
              flat_data1.append(color_histogram.flatten())
              # flat_data2.append(new_img)
              target.append(int(os.path.join(path,img)[51:53]))

          flat_data1 = np.array(flat_data1)
          # flat_data2 = np.array(flat_data2)
          target = np.array(target)

In [117]: print(len(flat_data1))
           print(len(flat_data2))
           print(len(target))

832
0
832
```

Hiển thị số lượng ảnh của từng loại bướm:



B5: Sử dụng mô hình học máy và thu kết quả.

Sử dụng ‘train_test_split’ để chia bộ dữ liệu thành bộ ‘train’ và bộ ‘test’.

Sử dụng mô hình học máy ‘RandomForestClassifier’ để huấn luyện bộ dữ liệu:

```
In [140]: from sklearn.ensemble import RandomForestClassifier
model1 = RandomForestClassifier()
model1.fit(x_train,y_train)
```

```
Out[140]: RandomForestClassifier
RandomForestClassifier()
```

```
In [141]: y_pred=model1.predict(x_test)
model1.score(x_test,y_test)
```

```
Out[141]: 0.572
```

Sử dụng mô hình ‘KneighborsClassifier’:

```
In [142]: from sklearn.neighbors import KNeighborsClassifier
model2 = KNeighborsClassifier()
model2.fit(x_train,y_train)
y_pred=model2.predict(x_test)
model2.score(x_test,y_test)
```

```
Out[142]: 0.208
```

Nhận xét:

- Độ chính xác của mô hình sẽ giảm dần nếu giảm số lượng màu lấy mẫu (64→16→8).
- Khi phân đoạn màu sắc trong ma trận tổng hợp 10 ảnh, khi lấy nhiều pixel ảnh cũng gây giảm độ chính xác (1500→2000→10000→20000).
- Do các bức ảnh có các vùng rộng xung quanh bướm nên cũng ảnh hưởng tới độ chính xác của mô hình nên cần sử dụng mặt nạ để tách phần chỉ chứa bướm.