Mã sinh viên : 20020688

Họ và tên : Đỗ Đức Mạnh

Bài thực hành số 6

Môn : Xử lý ảnh và thị giác robot

Bài tập: Sử dụng bộ dữ liệu có sẵn Digikala (Tên một hệ thống siêu thị lớn ở khu vực châu Á). Bộ dữ liệu này bao gồm các bức ảnh chụp các mặt hàng trên trang web với các màu sắc khác nhau. Trong bài thực hành này sinh viên sử dụng lược đồ histogram của kênh màu H trong HSV để làm vector nhận dạng. Hãy trích xuất đặc trưng cho tập ảnh, với mỗi ảnh thu được một vector histogram (H) và sử dụng nó để nhận dạng.

B1: Xác định đường dẫn đến từng ảnh trong bộ dữ liệu.

Sau khi tải bộ dữ liệu về, xác định đường dẫn trong từng tập ảnh lưu theo màu.

```
In [2]: data_path = './train'

In [3]: class_name = ['black','blue','brown','green','grey','orange','pink','purple','red','silver','white','yellow']
        color_number = 3
        def get_list_files(dirName):
            files_list = os.listdir(dirName)
            return files_list

        file_lists_color = get_list_files(data_path+'/'+class_name[color_number])
```

B2: Sử dụng hàm cắt ảnh và đo kênh H trong không gian màu HSV.

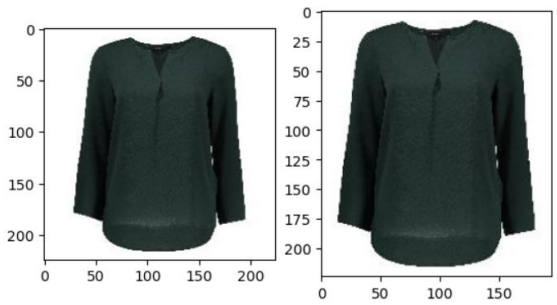
Kênh H được lưu dưới dạng ma trận 1 chiều có 256 phần tử.

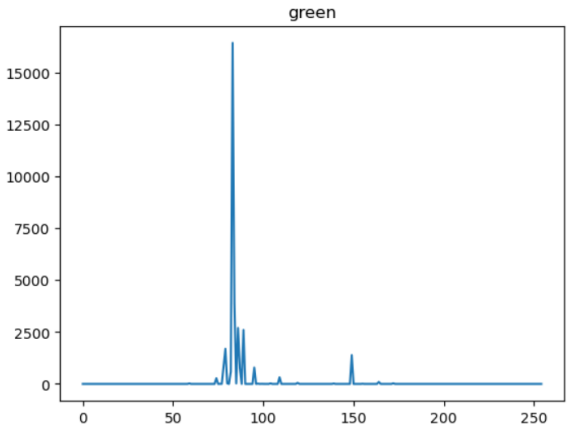
```
In [8]: def crop_img(img):
            mask = img!=255
            mask = mask.any(2)
            mask0, mask1 = mask.any(0), mask.any(1)
            colstart, colend = mask0.argmax(), len(mask0) - mask0[::-1].argmax()+1
            rowstart, rowend = mask1.argmax(), len(mask1) - mask1[::-1].argmax()+1
            return img[rowstart:rowend, colstart:colend]

In [9]: # a = crop_img(img)
        # plt.imshow(a)
        # plt.show()

In [10]: def hsv_histogram(img):
             hsv = cv2.cvtColor(img, cv2.COLOR_RGB2HSV)
         #     print(hsv)
             h = hsv[..., 0]
         #     print(h)
             return np.bincount(h.ravel(), minlength=256)
```

B3: In ra ảnh bất kì trong tập dữ liệu và kênh H của ảnh đó.

```
for i in range(1):
    fig,(ax1,ax2) = plt.subplots(1,2)
    rand_img_dir = rand_img(file_lists_color)
    print(rand_img_dir)
    imgBGR = cv2.imread(rand_img_dir)
    imgRGB = cv2.cvtColor(imgBGR,cv2.COLOR_BGR2RGB)
    img_crop = crop_img(imgRGB)
    ax1.imshow(imgRGB)
    ax2.imshow(img_crop)
    plt.show()

hist = hsv_histogram(img_crop)
plt.plot(hist[1:])
plt.title(class_name[color_number])
plt.show()
```

```
./train/green/117102353.jpg
```



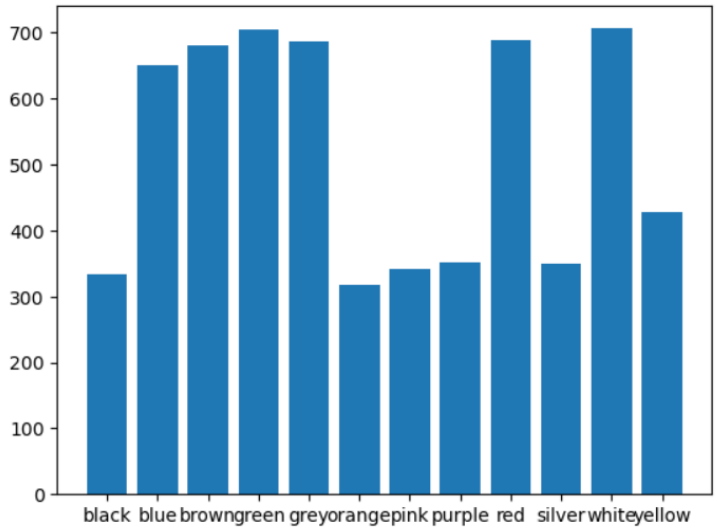


B4: Dán nhãn cho từng ảnh trong bộ dữ liệu.

Do các ảnh ở từng bộ màu khác nhau, chưa có ma trận lưu nhãn của từng ảnh đó để đưa vào các thuật toán học máy nên tạo 2 ma trận lưu ảnh và nhãn của ảnh tương ứng. Nhãn của ảnh là 12 màu được quy chuẩn về (0,11).

In [13]:
```
target = []
images = []
flat_data = []

for catagory in class_name:
#     print(catagory)
    class_num = class_name.index(catagory) #Label encoding value
#     print(class_num)
    path = os.path.join(data_path,catagory)
#     print(path)
    for img in os.listdir(path):
        imgBGR = cv2.imread(os.path.join(path,img))
#         print(img_arr.shape)
        imgRGB = cv2.cvtColor(imgBGR, cv2.COLOR_BGR2RGB)
        img_crop = crop_img(imgRGB)
        hist256 = hsv_histogram(img_crop)
        hist = hist256[1:]
        flat_data.append(hist.flatten())
        target.append(class_num)

flat_data = np.array(flat_data)
target = np.array(target)
```

Trực quan hóa số lượng ảnh trong bộ dữ liệu:



B5: Chia bộ dữ liệu ảnh cùng ảnh thành bộ huấn luyện và kiểm thử.

Sử dụng hàm ‘train_test_split’ từ sklearn.

```
In [17]: #split data into train and test
         from sklearn.model_selection import train_test_split
         x_train, x_test, y_train, y_test = train_test_split(flat_data,target,test_size=0.3)
```

B6: Sử dụng mô hình học máy để huấn luyện.

Sử dụng mô hình ‘RandomForestClassifier’:

```
In [19]: from sklearn.ensemble import RandomForestClassifier
         model = RandomForestClassifier()
         model.fit(x_train,y_train)
```

B7: Kiểm tra mô hình qua bộ kiểm thử và thu được độ chính xác 63,46%.

```
In [20]: y_pred = model.predict(x_test)

In [21]: model.score(x_test,y_test)
Out[21]: 0.6346153846153846
```

B8: So sánh với các mô hình khác.

```
In [26]: from sklearn.ensemble import BaggingClassifier
         from sklearn.neighbors import KNeighborsClassifier
         model2 = BaggingClassifier()
         model2.fit(x_train,y_train)
Out[26]: BaggingClassifier
         BaggingClassifier()

In [28]: y_pred2= model2.predict(x_test)

In [29]: model2.score(x_test,y_test)
Out[29]: 0.6079059829059829
```

In [30]:
```
model3 = KNeighborsClassifier()
model3.fit(x_train,y_train)
y_pred3=model3.predict(x_test)
model3.score(x_test,y_test)
```

Out[30]: 0.5587606837606838

In [31]:
```
from sklearn.ensemble import ExtraTreesClassifier
model4 = ExtraTreesClassifier()
model4.fit(x_train,y_train)
y_pred4 = model4.predict(x_test)
model4.score(x_test,y_test)
```

Out[31]: 0.6458333333333334

In [33]:
```
from sklearn.ensemble import GradientBoostingClassifier
model5 = GradientBoostingClassifier()
model5.fit(x_train,y_train)
y_pred5 = model5.predict(x_test)
model5.score(x_test,y_test)
```

Out[33]: 0.6228632478632479