

LẬP TRÌNH SONG SONG ỨNG DỤNG

BÁO CÁO TIẾN ĐỘ

PLANT LEAF DISEASE DETECTION USING XGBOOST



Nhóm Double Slash

20120165 - Hồng Nhất Phương

19120522 - Phạm Quốc Hưng

01

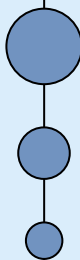
Khám phá dữ liệu

02

Trích xuất đặc trưng

03

Kế hoạch báo cáo tiếp theo



01

Khám phá dữ liệu



Khám phá dữ liệu

Tên bộ dữ liệu: Tomato Leaf Disease Detection

Nguồn dữ liệu: Kaggle

Số mẫu huấn luyện: 10.000 mẫu

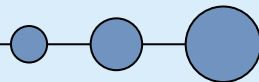
Số mẫu dự đoán: 1.000 mẫu

Khám phá dữ liệu

Số lớp: 10 lớp, bao gồm 1 lớp khỏe mạnh (healthy) và 9 lớp bệnh:

- Tomato_mosaic_virus
- Target_Spot
- Bacterial_spot
- Septoria_leaf_spot
- Late_blight
- Leaf_Mold
- Early_blight
- Spider_mites Two-spotted_spider_mite
- Tomato_Yellow_Leaf_Curl_Virus

Mỗi lớp có 1.000 mẫu



Bacterial spot



Early blight



Healthy



Late blight



Leaf mold



Septoria leaf spot



Spider mites



Target spot



Mosaic virus



Yellow leaf curl virus

02

Trích xuất đặc trưng



Trích xuất đặc trưng

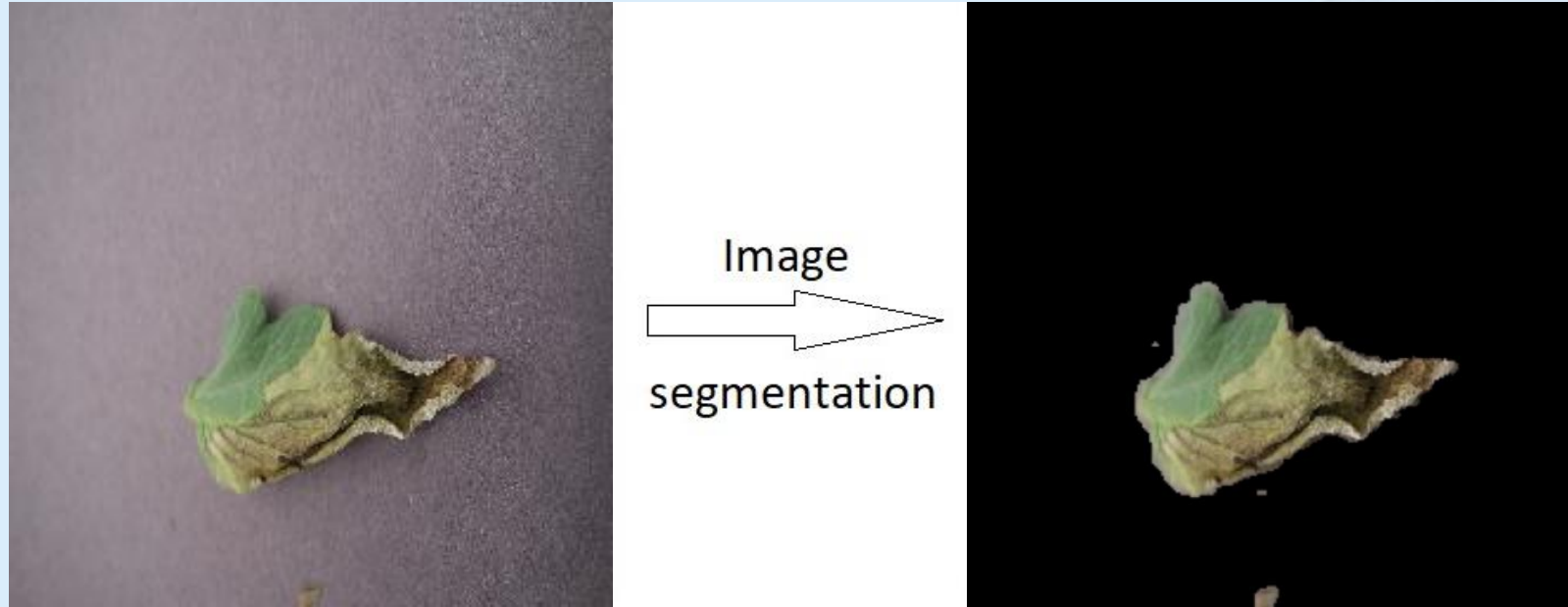
Rút trích đặc trưng tạo ra những đặc trưng tách lớp tốt cho việc phân lớp mẫu.

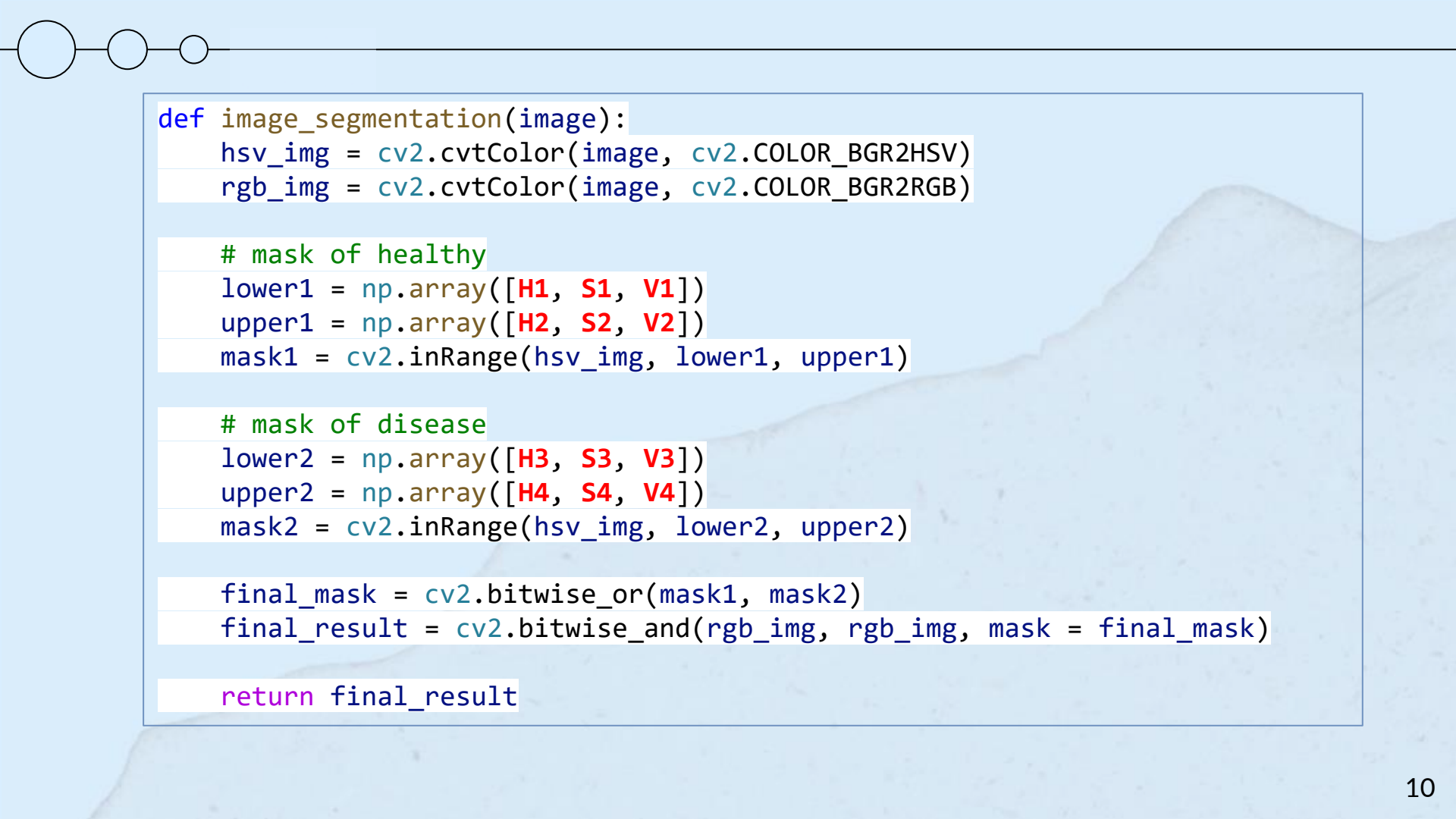
Quá trình thực hiện rút trích đặc trưng sẽ trải qua những bước sau:

- Phân đoạn ảnh dựa trên Color detection
- Rút trích đặc trưng dựa vào kết quả thu được ở trên
- Scale lại đặc trưng và sử dụng label encoder
- Lưu kết quả



Phân đoạn ảnh dựa trên Color detection là công việc tách ảnh chiếc lá ra khỏi nền dựa trên màu sắc, hình ảnh chiếc lá được lấy ra khỏi nền bằng cách sử dụng ranh giới màu hsv.



The slide features a light blue background with a faint mountain range. In the top-left corner, there are three white circles of varying sizes arranged horizontally. The Python code is presented in a light blue box with horizontal white highlights behind each line of code.

```
def image_segmentation(image):  
    hsv_img = cv2.cvtColor(image, cv2.COLOR_BGR2HSV)  
    rgb_img = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)  
  
    # mask of healthy  
    lower1 = np.array([H1, S1, V1])  
    upper1 = np.array([H2, S2, V2])  
    mask1 = cv2.inRange(hsv_img, lower1, upper1)  
  
    # mask of disease  
    lower2 = np.array([H3, S3, V3])  
    upper2 = np.array([H4, S4, V4])  
    mask2 = cv2.inRange(hsv_img, lower2, upper2)  
  
    final_mask = cv2.bitwise_or(mask1, mask2)  
    final_result = cv2.bitwise_and(rgb_img, rgb_img, mask = final_mask)  
  
    return final_result
```

2017/11/28 23:08:04 CST

(1) H-S (H: 0-180, S: 0-255, V: 255)

50

100

150

200

250

10 20 30 40 50 60 70 80 90 100 110 120 130 140 150 160 170 180

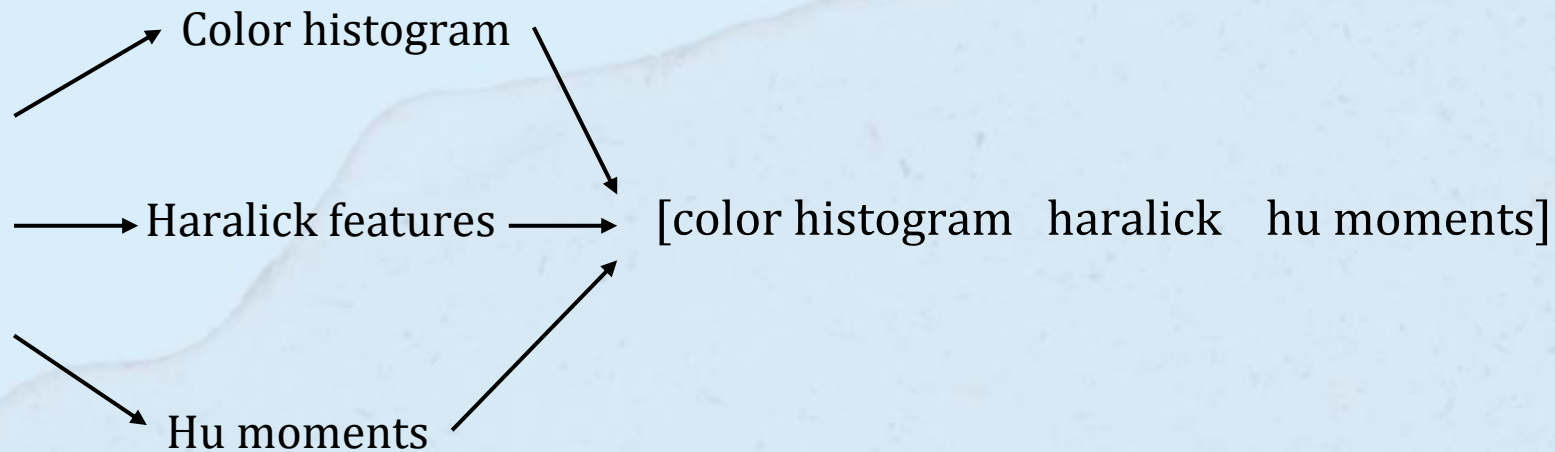
(2) H-S (H: 0-180, S: 255, V: 255)

知乎专栏 | python-opencv-image-processing

Source: Choose the correct upper and lower hsv boundaries

Sử dụng các Global feature descriptors để tiến hành trích xuất đặc trưng.

Global feature descriptors thường được sử dụng trong việc truy xuất hình ảnh, phát hiện và phân loại đối tượng. Các đặc trưng toàn cục (Global features) mô tả toàn bộ hình ảnh để khái quát hóa toàn bộ đối tượng, bao gồm biểu diễn màu sắc, mô tả hình dạng và đặc trưng kết cấu,...





Color histogram



```
cv2.calcHist()
```

Haralick features



```
mahotas.features.haralick()
```

Hu moments

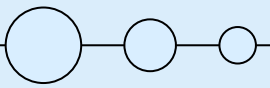


```
cv2.HuMoments(cv2.moments())
```



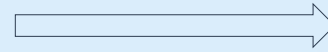
```
np.hstack([.....])
```

[color histogram haralick hu moments]



[color histogram haralick hu moments]
[color histogram haralick hu moments]
[color histogram haralick hu moments]
[color histogram haralick hu moments]
[color histogram haralick hu moments]
[color histogram haralick hu moments]

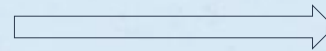
Min Max Scale



X

[healthy Leaf_mold Early_blight]

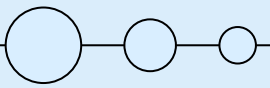
Label encoder



y

03

Kế hoạch báo cáo tiếp theo



STT	Tên công việc	
1	Thu thập dữ liệu	✓
2	Trích xuất đặc trưng	✓
3	Cài đặt XGBoost tuần tự	Báo cáo vào lần tiếp theo
4	Cài đặt XGBoost song song	
5	Áp dụng kỹ thuật cải tiến	



THE END