

Mã sinh viên : 20020688

Họ và tên : Đỗ Đức Mạnh

## Bài thực hành số 8

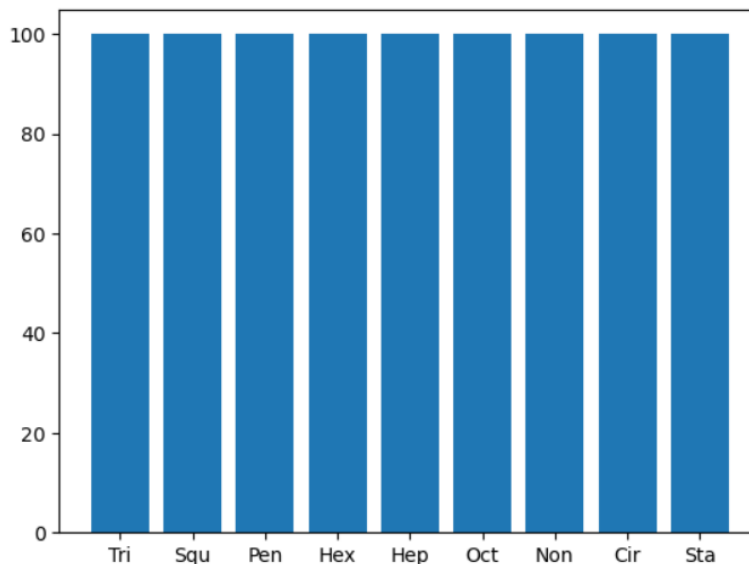
Môn : Xử lý ảnh và thị giác robot

Đề bài: Sử dụng đặc trưng HOG để thực hiện bài toán nhận dạng đối tượng và xác định vị trí của đối tượng đó trong ảnh với dữ liệu “2D geometric shapes dataset.”

### Bài làm

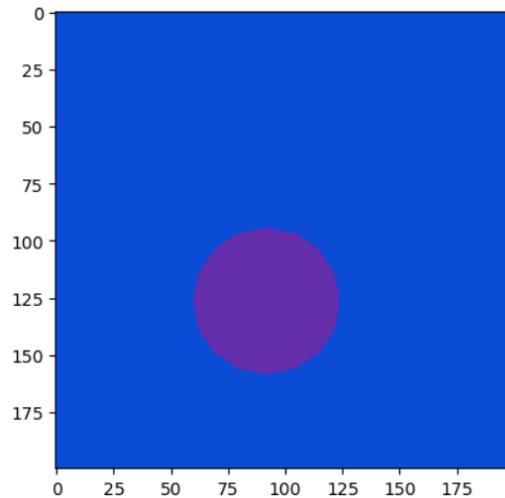
#### B1: Kiểm tra bộ dữ liệu cho trước

- Bộ dữ liệu có 90000 ảnh gồm 9 loại ảnh về các hình học: ‘triangle’, ‘square’, ‘pentagon’, ...
- Do số lượng lớn nên em sẽ tách thành bộ dữ liệu mới với kích thước 900 ảnh và mỗi loại có 100 ảnh.



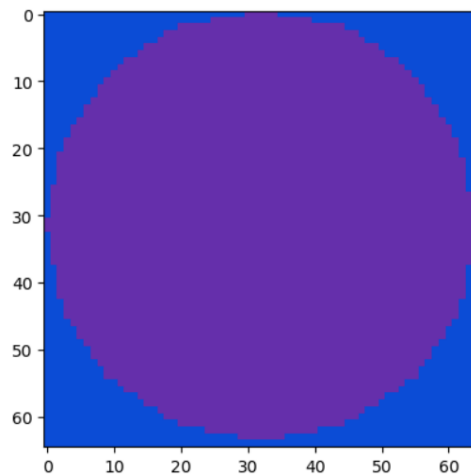
#### B2: Trích xuất đặc trưng HOG

- Ta đọc được 1 ảnh gốc bất kì trong class ‘Circle’:



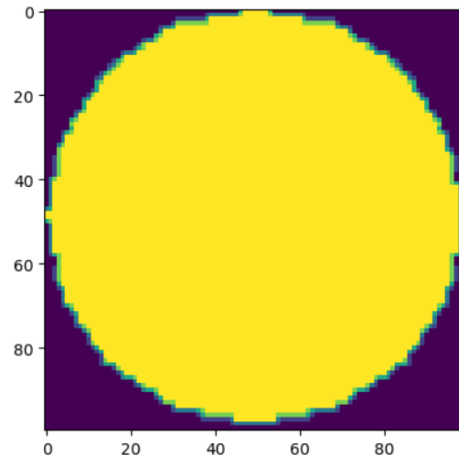
*Ảnh gốc*

- Nếu đưa ảnh có kích thước 200x200 vào mô hình học máy sẽ không tối ưu được thời gian nên ta sẽ cắt phần ảnh chứa hình tròn qua hàm ‘crop\_img’ trong Bài sử dụng HSV.



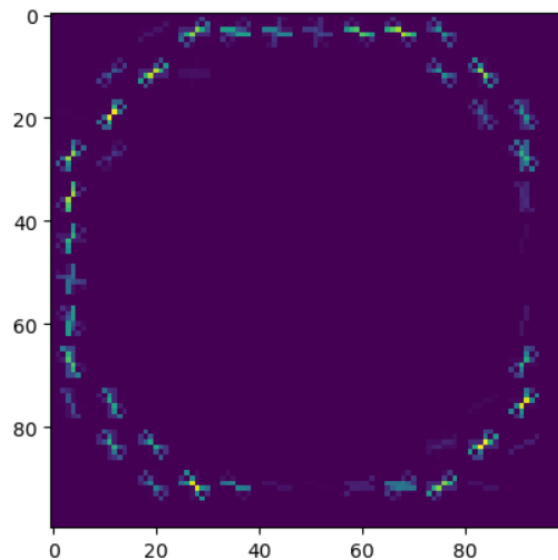
*Ảnh cắt*

Trước khi tách đặc trưng HOG cần chuyển thành ảnh đen trắng, đồng thời chỉnh kích thước về (100x100) để các ảnh cắt về cùng kích thước.



*Ảnh cắt gray và resize (100x100)*

- Sau đó tách đặc trưng HOG bằng hàm có sẵn trong thư viện 'skimage' với các tham số là 8 hướng, chia ảnh thành các khối pixel có kích thước (8x8), và chuẩn hóa (2x2) khối theo norm bậc 2 để đưa các ảnh về cùng một vecto biểu diễn do khác biệt về histogram. Ta sẽ thu được bức ảnh đặc trưng hog cùng vecto biểu diễn.



*Ảnh HOG*

B3: Sử dụng mô hình học máy để phân loại ảnh.

Vecto biểu diễn đặc trưng HOG của từng ảnh sẽ được đưa vào mô hình học máy cùng nhãn của bức ảnh đó.

Sử dụng mô hình ‘RandomForestClassifier’ thu được độ chính xác 97.04%. Mô hình ‘KNeighborsClassifier’ có độ chính xác 98.89%.

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

: RandomForestClassifier()
In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

: y_pred=model1.predict(x_test)
model1.score(x_test,y_test)

: 0.9703703703703703

: 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)

: 0.9888888888888889
```

## **Nhận xét:**

Khi thay đổi số chiều của đặc trưng HOG (2→4→8) độ chính xác thay đổi từ 97.78 →96.30→96.67 sử dụng ‘RandomForestClassifier’. Sự thay đổi về độ chính xác không nhiều do ảnh đã được cắt sát hình khối, màu hình khối và màu nền có sự khác biệt nên độ chính xác rất cao. Tuy nhiên khi tăng các hướng tách đặc trưng HOG có thể xảy ra hiện tượng trùng hướng của cạnh các hình đa giác giữa class ‘octagon’ và ‘nontagon’.

Link code: <https://github.com/dducmanh99/ImageProcessing.git>