

**BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC NHA TRANG
KHOA CÔNG NGHỆ THÔNG TIN**



BÁO CÁO BÀI TẬP NHÓM

**TÌM HIỂU PHƯƠNG PHÁP TRÍCH CHỌN ĐẶC
TRÙNG GLCM**

Giảng viên hướng dẫn: TS. Đinh Đồng Lương

Sinh viên thực hiện: Cao Minh Tiến (nhóm trưởng)

Phùng Sỹ Hoàng Sơn

Dương Tư Hiệu (thư kí)

Nguyễn Hữu Thành

Nguyễn Mai Diễm Hương

Nguyễn Quốc Bảo

Hoàng Thái Duy

Khánh Hòa – 2022

**BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC NHA TRANG
KHOA CÔNG NGHỆ THÔNG TIN**



**TÌM HIỂU PHƯƠNG PHÁP TRÍCH CHỌN ĐẶC
TRÙNG GLCM**

Giảng viên hướng dẫn: TS. Đinh Đồng Lương

Sinh viên thực hiện: Cao Minh Tiến (nhóm trưởng)

Phùng Sỹ Hoàng Sơn

Dương Tư Hiệu (thư kí)

Nguyễn Hữu Thành

Nguyễn Mai Diễm Hương

Nguyễn Quốc Bảo

Hoàng Thái Duy

Khánh Hòa – 2022

DANH SÁCH PHÂN CÔNG NHIỆM VỤ

Nội dung	Thành viên đảm nhận
Giới thiệu về texture feature và các ứng dụng của nó	Phùng Sỹ Hoàng Sơn
Giới thiệu về GLCM	Nguyễn Quốc Bảo, Phùng Sỹ Hoàng Sơn, Nguyễn Mai Diễm Hương, Hoàng Thái Duy
Chi tiết GLCM	Nguyễn Hữu Thành, Dương Tư Hiệu, Cao Minh Tiến
Ví dụ thực tế	Cao Minh Tiến, Nguyễn Mai Diễm Hương

Giao nhiệm vụ, tổng hợp và biên soạn báo cáo: Cao Minh Tiến

Đánh giá của nhóm trưởng đối với từng thành viên

Tên thành viên	Đánh giá
Phùng Sỹ Hoàng Sơn	A+, hoàn thành rất tốt công việc được giao
Nguyễn Mai Diễm Hương	A, còn vài mục nhỏ chưa hoàn thành
Dương Tư Hiệu	A, còn vài mục chưa hoàn thành
Nguyễn Hữu Thành	A, còn vài mục chưa hoàn thành
Nguyễn Quốc Bảo	B, chưa hoàn thành tốt công việc được giao
Hoàng Thái Duy	B-, ít năng nổ trong hoạt động nhóm và công việc, chưa hoàn thành tốt công việc được giao

MỤC LỤC

DANH SÁCH PHÂN CÔNG NHIỆM VỤ	3
MỤC LỤC	4
CHƯƠNG 1: TỔNG QUAN NGHIÊN CỨU	6
1.1. Thuật toán trích chọn đặc trưng	6
1.2. Texture feature (đặc trưng kết cấu)	6
1.2.1. Giới thiệu	6
1.2.2. Ứng dụng	7
1.3. GLCM	7
1.3.1. Lịch sử phát triển	7
1.3.2. Chi tiết về GLCM	8
1.3.2.1. GLCM là gì.....	8
1.3.2.2. Các thuộc tính thống kê của hình ảnh có nguồn gốc từ GLCM.....	8
1.3.2.3. Thuật toán	10
1.3.3. Ưu, nhược điểm của thuật toán GLCM	10
1.3.3.1. Ưu điểm	10
1.3.3.2. Nhược điểm	10
1.4. Bài toán phân loại trong học máy.....	11
1.5. Công cụ sử dụng.....	11
1.5.1. Visual Studio Code.....	11
1.5.2. Jupyter Notebook.....	11
1.5.2.1. Jupyter là gì?.....	11
1.5.2.2. Jupyter Notebook là gì?.....	12
CHƯƠNG 2: DEMO THUẬT TOÁN GLCM VÀ BÀI TOÁN PHÂN LOẠI HÌNH ẢNH	13

2.1. Demo thuật toán GLCM	13
2.2. Bài toán phân loại hình ảnh.....	17
CHƯƠNG 3: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN.....	18
TÀI LIỆU THAM KHẢO.....	19

CHƯƠNG 1: TỔNG QUAN NGHIÊN CỨU

1.1. Thuật toán trích chọn đặc trưng

– Trích chọn đặc trưng là một trong những thuật toán cốt lõi trong các bài toán về nhận diện (recognition), phát hiện (detection), phân loại (classification), phân cụm (clustering),... trong học máy. Đây thực chất là một quá trình chọn lọc một tập con chứa các thuộc tính liên quan để sử dụng trong quá trình xây dựng mô hình. Các kỹ thuật trích chọn đặc trưng được dùng cho một số lý do:

- + Đơn giản hóa các mô hình để giúp các nhà nghiên cứu/người dùng diễn dịch dễ dàng hơn
- + Giảm thời gian huấn luyện
- + Giảm chiều dữ liệu
- + Tăng cường tổng quát hóa bằng cách giảm sự quá khớp (overfitting)

Các đặc trưng thường gặp trong các bài toán học máy có thể được chia làm 4 nhóm chính, bao gồm: Geometric features (các đặc trưng hình học), Statistical features (các đặc trưng thống kê), Texture features (các đặc trưng kết cấu) ,and Color features (các đặc trưng màu sắc)

1.2. Texture feature (đặc trưng kết cấu)

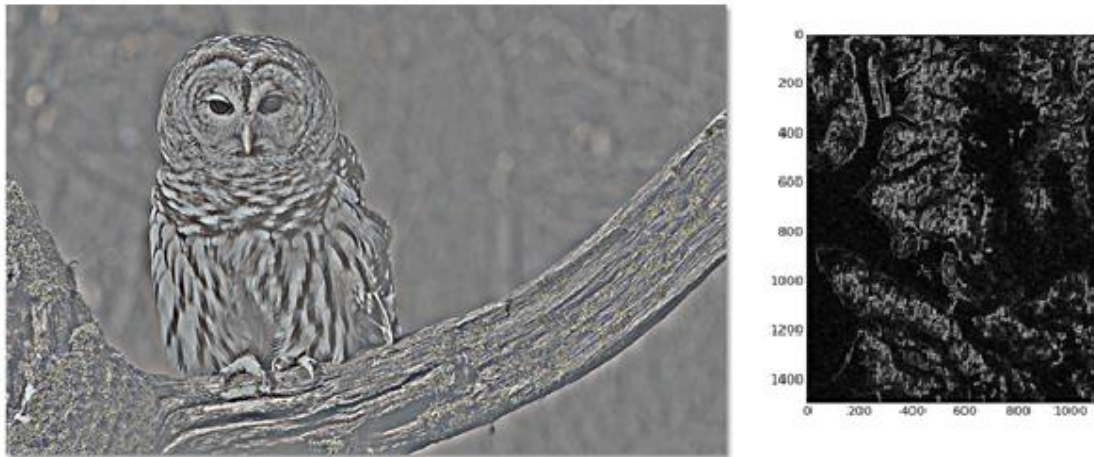
1.2.1. Giới thiệu

Texture là đặc điểm quan trọng nhất đối với nhiều loại hình ảnh xuất hiện ở khắp mọi nơi trong tự nhiên chẳng hạn như hình ảnh y tế và hình ảnh cảm biến, v.v. Texture được định nghĩa là bề mặt biểu hiện của hệ thống thị giác của con người về các đối tượng tự nhiên. Mọi người đều dễ dàng nhận ra, nhưng rất khó để xác định kết cấu trong ma trận, nhưng nó vẫn xảy ra trong một khu vực của ma trận đã phân tích dựa vào định lượng và định tính.

Việc trích xuất đối tượng với phân tích texture được thực hiện bằng cách lấy các đối tượng từ ảnh thang độ xám ở dạng entropy, contrast, energy, homogeneity, gray scale, và standard deviation, trong khi trích xuất đối tượng từ ảnh màu là red (R), green (G), and blue (B).

1.2.2. Ứng dụng

Giúp con người có thể phát hiện vùng đặc trưng ảnh với thời gian nhanh chóng, kích thước file nhỏ và không chứa quá nhiều tham số. Việc trích chọn đặc trưng này hỗ trợ rất nhiều trong các vấn đề thực tiễn như : Ảnh vệ tinh, thám hiểm, nhiếp ảnh, phát hiện và phân loại đối tượng video theo thời gian thực,....



1.3. GLCM

1.3.1. Lịch sử phát triển

- Vào khoảng những năm thập niên 60 - 70 của Thế kỉ 20 nhu cầu sử dụng các thông tin từ hình ảnh số tăng cao. Đặc biệt là: giới làm tin . Tuy nhiên hệ thống lưu trữ hình ảnh bằng nguyên lý quang học (máy film) gặp nhiều vấn đề trở ngại và có rủi ro cao. Ý tưởng số hoá ảnh bằng máy scan và ý tưởng số hoá tín hiệu ảnh động xuất hiện trước ý tưởng chụp ảnh số đánh dấu bước chuyển mình quan trọng trong lĩnh vực hình ảnh.
- Phân tích các đặc trưng cho sự biến đổi không gian của hình ảnh dựa trên một số thủ tục và mô hình toán học để trích xuất thông tin từ nó. Một trong những phương pháp sớm nhất được đề xuất bởi Robert Haralick vào những năm 1973, được biết đến với tên gọi Ma trận đồng xuất hiện mức xám (GLCM) và kể từ đó biểu đồ ma trận GLCM hai chiều này được sử dụng rộng rãi trong “ phân tích kết cấu ”.
- Phương pháp được Robert Haralick giới thiệu đã áp dụng rất thành công trên nhiều lĩnh vực cần đến phân tích kết cấu của hình ảnh. Điển hình như Google đã cho ra mắt Google Maps vào khoảng tháng 2 năm 2005 dựa trên việc sử dụng ảnh chụp từ vệ tinh để phân tích kết cấu và xây dựng mô phỏng lại Trái Đất.

– Không chỉ áp dụng được trên ảnh chụp vệ tinh Trái Đất, mà nó còn phục vụ rất cần thiết vào ngành khám phá vũ trụ ngày nay.

1.3.2. Chi tiết về GLCM

1.3.2.1. GLCM là gì

GLCM là viết tắt của Gray Level Co-occurrence Matrix (tạm dịch: Ma trận đồng xuất hiện mức xám). Đây là một ma trận chứa đựng số lần xuất hiện của 2 mức xám trong tất cả các cặp mức xám của một hình ảnh và được sử dụng cho việc trích xuất đặc trưng từ hình ảnh. Cột và hàng của GLCM là giá trị mức xám của điểm ảnh đang xét và điểm ảnh đáp ứng theo không gian và khoảng cách tương ứng. Do đó, kích thước của GLCM bằng với giá trị mức xám được biểu diễn của hình ảnh. GLCM có 2 thông số để tính toán ma trận là khoảng cách giữa các điểm ảnh d và góc điểm ảnh lân cận.

1.3.2.2. Các thuộc tính thống kê của hình ảnh có nguồn gốc từ GLCM

Đặc trưng	Mô tả	Công thức
Contrast	<ul style="list-style-type: none"> • Trả về một thước đo độ tương phản cường độ giữa một pixel và các pixel lân cận của nó trên toàn bộ hình ảnh. • Độ tương phản = 0 cho một hình ảnh không đổi. 	$\sum_{i,j}^{N-1} P_{i,j} (i - j)^2$
Correlation	<ul style="list-style-type: none"> • Trả về một thước đo về mức độ tương quan một pixel với pixel lân cận của nó trên toàn bộ hình ảnh. • Range = [-1 1] • Tương quan là -1 hoặc 1 đối với hình ảnh không tương quan hoặc tương quan. Tương quan là NaN cho một ảnh không đổi. 	$\sum_{i,j}^{N-1} P_{i,j} \left[\frac{(i - \mu_i)(j - \mu_j)}{\sigma_i \sigma_j} \right]$
Energy	<ul style="list-style-type: none"> • Trả về căn bậc 2 của ASM, (trong đó ASM cũng là một trong 14 đặc trưng kết cấu được giới thiệu bởi Haralick), với 	\sqrt{ASM} <p>với $ASM = \sum_{i,j=0}^{N-1} P_{i,j}^2$</p>

	ASM bằng tổng các phần tử bình phương trong GLCM <ul style="list-style-type: none"> • Energy = 1 thì hình ảnh không đổi. • Energy còn được gọi là tính đồng nhất, tính đồng nhất của Energy và mômen góc thứ hai. 	
Homogeneity	<ul style="list-style-type: none"> • Trả về 1 giá trị đo lường cách nó gần ra sao so với phân bố của phần tử trong đường chéo ma trận GLCM • Range = [0 1] • Tính đồng nhất = 1 đối với diagonal GLCM 	$\sum_{i,j=0}^{N-1} \frac{P_{i,j}}{1 + i - j }$
Entropy	Entropy chỉ định độ không chắc chắn, ngẫu nhiên trong các giá trị hình ảnh. Đo lượng thông tin trung bình cần thiết để mã hóa các giá trị hình ảnh.	$- \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} P_{i,j} \log(P_{i,j})$

Trong đó:

+ $P_{i,j}$ chính là giá trị của phần tử tại hàng thứ i, cột thứ j của ma trận GLCM đối xứng chuẩn hóa (cách tính ma trận này sẽ được trình bày ở phần thuật toán)

+ N: Biểu thị số mức xám trong hình ảnh. $N >$ giá trị cường độ mức xám lớn nhất của hình ảnh đa mức xám (trong nhiều công thức, N được thay bằng Level)

+ μ : giá trị trung bình của ma trận GLCM (ước lượng các cường độ mức xám của tất cả các pixel), công thức:

$$\mu = \sum_{i,j=0}^{N-1} i P_{i,j}$$

+ σ^2 : Phương các giá trị cường độ mức xám, công thức:

$$\sigma^2 = \sum_{i,j=0}^{N-1} P_{i,j} (i - \mu)^2$$

1.3.2.3. Thuật toán

❖ **Bước 1:** chuyển đổi ảnh màu sang ảnh đa mức xám, đồng thời ‘resize’ các hình ảnh về cùng một kích thước (chẳng hạn 128x128)

❖ **Bước 2:** Tính toán ma trận GLCM (lưu ý GLCM là một ma trận vuông có kích thước NxN, với N > giá trị cường độ mức xám lớn nhất của hình ảnh đã xử lý ở Bước 1)

- **Bước 2.1:** Tính toán ma trận GLCM đối xứng (bằng cách cộng ma trận GLCM với ma trận chuyển vị của nó)
- **Bước 2.2:** Chuẩn hóa ma trận GLCM (bằng cách chia mỗi phần tử cho tổng tất cả các phần tử). Mỗi phần tử của GLCM lúc này có thể xem như là xác suất của việc tìm ra mối quan hệ của 2 mức cường độ xám (i,j)

Bước 3: Tính toán các đặc trưng kết cấu. Lưu ý: sử dụng ma trận GLCM đối xứng chuẩn hóa.

1.3.3. Ưu, nhược điểm của thuật toán GLCM

1.3.3.1. Ưu điểm

- Tiêu hao ít tài nguyên
- Tổ chức dữ liệu gọn
- Dễ dàng chọn lọc
- File lưu trữ nhỏ
- Thực thi nhanh chóng, trong quá trình làm việc không chiếm dụng nhiều không gian là một lợi thế để có thể hoạt động tối ưu trên đa nền tảng □ Giúp người dùng có một cái nhìn trực quan

1.3.3.2. Nhược điểm

- Những đặc điểm của ma trận Co-occurent có những khó khăn như : không có một phương pháp được thiết lập hoàn hảo nào về sự chọn véc-tơ thay thế cho d và việc tính toán ma tranajin đối với một số giá trị khác nhau của d là không thể thực hiện được. Hơn nữa, với một giá trị của d có một số lượng lớn có thể được tính toán. Điều này có nghĩa là một số phương pháp lựa chọn đặc điểm có thể tính toán. Điều này có nghĩ là phương

pháp lựa chọn đặc điểm cần phải được sử dụng để lựa chọn những đặc điểm có liên quan nhất.

- Khó khôi phục về lại nguyên gốc sau khi đã thực hiện việc trích chọn
- Tìm phương pháp trích chọn đặc trưng phù hợp là phải tìm một tập hợp các tính năng kết cấu có khả năng phân biệt tốt. Hầu hết các tính năng kết cấu thường thu được từ việc áp dụng một toán tử cục bộ, phân tích thống kê hoặc đo lường trong một miền đã biến đổi.

1.4. Bài toán phân loại trong học máy

Đây là một trong những bài toán điển hình của học có giám sát (Supervised Learning). Một bài toán được gọi là phân loại (Classification) nếu các nhãn (label) của dữ liệu đầu vào được chia thành một số hữu hạn nhóm. Bộ lọc thư rác là ví dụ tiêu biểu cho tác vụ này: nó được huấn luyện với nhiều mẫu thư điện tử cùng với nhãn tương ứng (thư rác hoặc thư thông thường), từ đó học cách phân loại các thư điện tử mới.

Ví dụ: Gmail xác định xem một email có phải là spam hay không; các hãng tín dụng xác định xem một khách hàng có khả năng thanh toán nợ hay không. Ba ví dụ phía trên được chia vào loại này

1.5. Công cụ sử dụng

1.5.1. Visual Studio Code

+ Là một trình biên tập lập trình code miễn phí dành cho Windows, Linux và macOS, Visual Studio Code được phát triển bởi Microsoft. Nó được xem là một sự kết hợp hoàn hảo giữa IDE và Code Editor

+ Visual Studio Code hỗ trợ chức năng debug, đi kèm với Git, có syntax highlighting, tự hoàn thành mã thông minh, snippets, và cải tiến mã nguồn. Nhờ tính năng tùy chỉnh, Visual Studio Code cũng cho phép người dùng thay đổi theme, phím tắt, và các tùy chọn khác

1.5.2. Jupyter Notebook

1.5.2.1. Jupyter là gì?

+ Jupyter là một nền tảng tính toán khoa học mã nguồn mở, với khả năng nổi bật cho phép tương tác trực tiếp với từng dòng code (interactive), hỗ trợ hơn 40 ngôn ngữ lập trình, trong đó tập trung vào 3 ngôn ngữ là Julia, Python và R, cái tên Jupyter bắt nguồn

từ cách chơi chữ I Python, You R, We Julia, hay một công thức ngắn gọn Jupyter = Julia + Python + R. Bên cạnh đó, Jupyter cũng là một công cụ hoàn toàn miễn phí, được tạo ra với mục đích nhắm đến khoa học dữ liệu và giáo dục, giúp mọi người cùng học lập trình dễ dàng hơn (cụ thể ở đây là Python). Jupyter có tính tương tác nên có thể sử dụng làm môi trường chạy thử và giảng dạy.

1.5.2.2. Jupyter Notebook là gì?

+ Trước đây là có tên là IPython Notebook, đến năm 2014 được đổi thành Jupyter Notebook. Jupyter hỗ trợ rất nhiều các kernel (nhân) cho các ngôn ngữ khác nhau, khoảng trên 40 ngôn ngữ trong đó có Python. Việc đổi tên từ IPython sang Jupyter cũng là vì mục đích hỗ trợ đa ngôn ngữ.

+ Phần cơ bản của nó là một ứng dụng chạy trên nền web cho phép chạy Interactive Python (hay IPython), bạn có thể đưa cả code Python và các thành phần văn bản phức tạp như hình ảnh, công thức, video, biểu thức... vào trong cùng một file giúp cho việc trình bày trở lên dễ hiểu, giống như một file trình chiếu nhưng lại có thể thực hiện chạy code tương tác trên đó, cốt lõi của việc này chính là Markdown. Các file "notebook" này có thể được chia sẻ với mọi người và có thể thực hiện lại các công đoạn một cách nhanh chóng và chính xác như những gì bạn đã làm trong quá trình tạo ra file.

CHƯƠNG 2: DEMO THUẬT TOÁN GLCM VÀ BÀI TOÁN PHÂN LOẠI HÌNH ẢNH

2.1. Demo thuật toán GLCM

a. Tạo ma trận đồng xuất hiện mức xám

- + Nhập hình ảnh
- + Chuyển đổi hình ảnh thành ảnh đa mức xám
- + Sử dụng hàm *graycomatrix* trong thư viện *skimage.feature* của python để tạo ma trận glcm:

```
(function) graycomatrix: (image: Any, distances: Any, angles: Any, levels: Any
| None = None, symmetric: bool = False, normed: bool = False) -> (Any |
NDArray[bool_] | ndarray)
```

Trong đó:

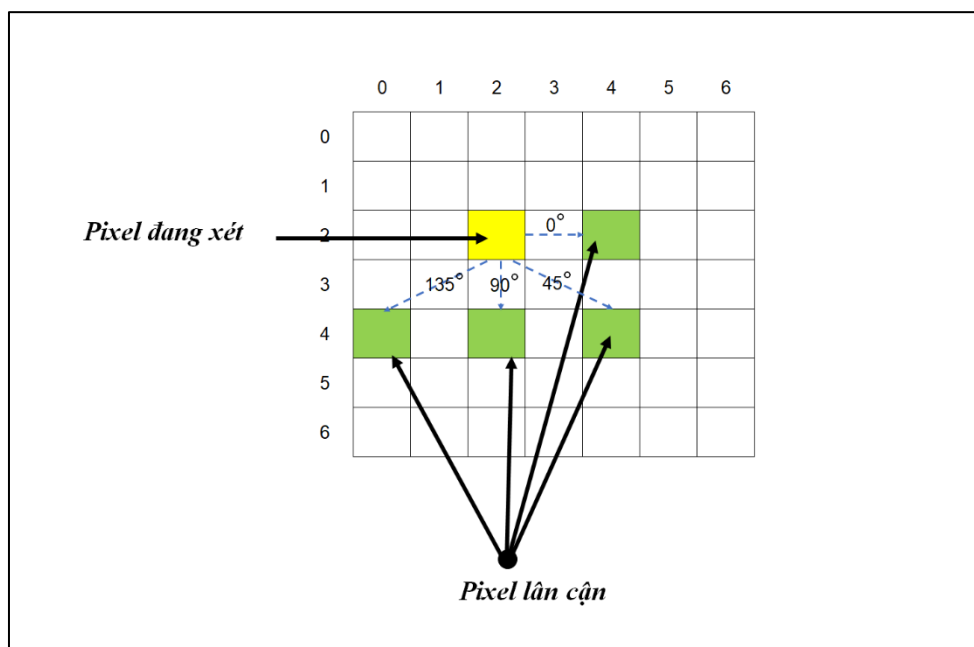
image: dữ liệu hình ảnh đầu vào (ảnh đa mức xám)

distances: khoảng cách giữa pixel đang xét với các pixel lân cận với nó (neighbors), có thể chỉ định nhiều khoảng cách khác nhau bằng cách truyền vào một list. Ví dụ:

```
graycomatrix(..., distances=[1,2,3],...)
```

angles: số đo góc hợp bởi pixel đang xét và pixel lân cận của nó cùng với đường nằm ngang. Có thể chỉ định nhiều số đo góc khác nhau bằng cách truyền vào một list.

```
graycomatrix(..., angles=[0,np.pi/2,np.pi/4],...)
```



levels: quy định kích thước của ma trận GLCM. Lưu ý: levels > giá trị mức xám cao nhất trong hình ảnh đầu vào

- Ngoài ra còn có các tham số khác như symmetric (tạo ma trận GLCM đối xứng, normed (chuẩn hóa ma trận GLCM), v.v..

+ Code minh họa:

```
import numpy as np
from skimage.feature import graycomatrix

img=np.array([[1,1,5,6,8],[2,3,5,7,1],[4,5,7,1,2],[8,5,1,2,5]],dtype='uint8')
glcm=graycomatrix(image=img,distances=[1],angles=[0],levels=9)
```

	0	1	2	3	4	5	6	7	8
0	0	0	0	0	0	0	0	0	0
1	0	1	2	0	0	1	0	0	0
2	0	0	0	1	0	1	0	0	0
3	0	0	0	0	0	1	0	0	0
4	0	0	0	0	0	1	0	0	0
5	0	1	0	0	0	0	1	2	0
6	0	0	0	0	0	0	0	0	1
7	0	2	0	0	0	0	0	0	0
8	0	0	0	0	0	1	0	0	0

b. Tính toán 5 giá trị đặc trưng mô tả các kết cấu của hình ảnh

**Các tham số truyền vào các hàm được viết dưới đây là mảng 2 chiều*

• Entropy

```
def cal_entropy(glcm):
    s_glcm=glcm+glcm.T
    n_glcm=s_glcm/np.sum(s_glcm)
    entropy_matrix=-np.log(n_glcm+1e-6)
    return np.sum(entropy_matrix*n_glcm)
```

• Contrast

```
def cal_contrast(glcm):
    s_glcm=glcm+glcm.T
    n_glcm=s_glcm/np.sum(s_glcm)
```

```

contrast=0
n=glcm.shape[0]
for i in range(n):
    for j in range(n):
        contrast=contrast+n_glcm[i][j]*(i-j)**2

return contrast

```

- Correlation

```

def cal_correlation(glcm):
    mu=np.array([0,0])
    sig=np.array([0,0])
    s_glcm=glcm+glcm.T
    n_glcm=s_glcm/np.sum(s_glcm)
    n=n_glcm.shape[0]
    for i in range(n):
        for j in range(n):
            mu=mu+np.array([i*n_glcm[i,j],j*n_glcm[i,j]])

    for i in range(n):
        for j in range(n):
            sig=sig+n_glcm[i,j]*((i-mu[0])**2)

    corr=0
    for i in range(n):
        for j in range(n):
            corr=corr+n_glcm[i,j]*(i-mu[0])*(j-mu[1])/np.sqrt(sig[0]*sig[1])

    return corr

```

- Energy

```

def cal_energy(glcm):
    s_glcm=glcm+glcm.T
    n_glcm=s_glcm/np.sum(s_glcm)
    energy=np.sum(n_glcm**2)
    return np.sqrt(energy)

```

- Homogeneity

```
def cal_homogeneity(glc):
    s_glc=glc+glc.T
    n_glc=s_glc/np.sum(s_glc)
    n=glc.shape[0]
    homo=0
    for i in range(n):
        for j in range(n):
            homo=homo+n_glc[i][j]/(1+(i-j)**2)

    return homo
```

- Trong python có cung cấp hàm *graycoprops* để tính các giá trị đặc trưng kết cấu (nhưng không có entropy)

```
contrast = graycoprops(glc,'contrast')
correlation = graycoprops(glc,'correlation')
energy = graycoprops(glc,'energy')
homogeneity = graycoprops(glc,'homogeneity')
```

- Code minh họa thuật toán trích chọn đặc trưng:

```
import cv2
import pandas as pd
import numpy as np
from skimage.feature import graycomatrix,graycoprops
SIZE_IMAGE=128

def cal_entropy(glc):
    s_glc=glc+glc.T
    n_glc=s_glc/np.sum(s_glc)
    entropy_matrix=-np.log(n_glc+1e-6)
    return np.sum(entropy_matrix*n_glc)

img=cv2.imread('natural_images/test/airplane/airplane_0701.jpg',0)
img=cv2.resize(img,(SIZE_IMAGE,SIZE_IMAGE))
glc=graycomatrix(img,[1],[0])
print('Kích thước ma trận hình ảnh: {}'.format(img.shape[0],img.shape[1]))
df = pd.DataFrame()
df['entropy'] = [cal_entropy(glc[:, :, 0, 0])]
df['contrast'] = graycoprops(glc,'contrast')[0]
df['correlation'] = graycoprops(glc,'correlation')[0]
df['energy'] = graycoprops(glc,'energy')[0]
```



```
df['homogeneity'] = graycoprops(g lcm, 'homogeneity')[0]
print('Các giá trị đặc trưng kết cấu: ')
print(df)
Kích thước ma trận hình ảnh: 128x128
Các giá trị đặc trưng kết cấu:
      entropy  contrast  correlation  energy  homogeneity
0  7.880041  336.714013    0.94475  0.044428    0.258599
```

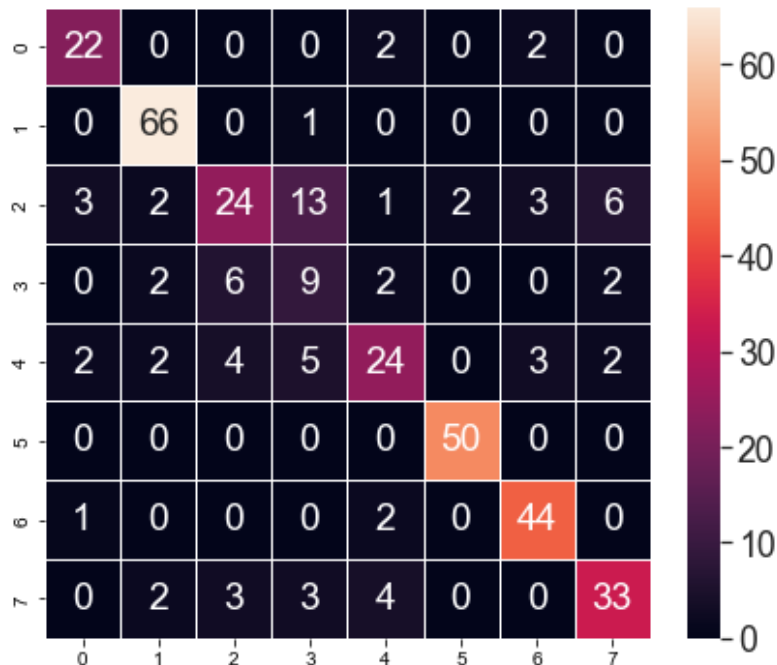
2.2. Bài toán phân loại hình ảnh

Tập dữ liệu: dữ liệu [natural](#) trên Kaggle, bao gồm 8 lớp phân biệt: airplane, car, cat, dog, flower, fruit, motorbike, person

Thuật toán học tập: Light Gradient Boosting Machine (dựa trên thuật toán cây quyết định, thường được sử dụng để xếp hạng, phân loại và nhiều tác vụ học máy khác). Đây là một trong những thuật toán rất phổ biến hiện nay nhờ vào hiệu suất tuyệt vời của nó

Thuật toán trích chọn đặc trưng được sử dụng: GLCM, và từ ma trận GLCM trích xuất ra 5 đặc trưng là entropy, contrast, correlation, energy, homogeneity

Kết quả: độ chính xác đạt được là 77.3% trên tập dữ liệu kiểm tra



Hình 1.1 Các giá trị trên đường chéo chính biểu thị các trường hợp dự đoán đúng

Source Code: <https://github.com/iamcaominhtien/mid-term-ml/tree/master/project>

CHƯƠNG 3: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

Sau 2 tuần nghiên cứu và thực hiện đề tài, nhóm chúng em đã hoàn thành bài tập lớn giữa kì với chủ đề “Tìm hiểu phương pháp trích chọn đặc trưng GLCM”. Đề cương đã giới thiệu một cách tổng quan nhất về ma trận GLCM, cùng với các đặc trưng kết cấu của hình ảnh được tính toán từ ma trận GLCM. Bằng cách sử dụng 2 công cụ là Visual Studio Code và Jupyter Notebook, nhóm chúng em đã demo được thuật toán trích chọn đặc trưng GLCM, cũng như chạy được một project học máy liên quan đến phân loại hình ảnh có sử dụng phương pháp trích chọn đặc trưng GLCM, từ đó thì nhóm chúng em đã thu được những hiểu biết sâu sắc hơn về đề tài nhóm mình thực hiện, cũng như cách khởi chạy một project học máy từ đầu đến cuối

Mặc dù đã hoàn thiện, song project học máy áp dụng phương pháp GLCM vẫn còn nhiều thiếu sót, chưa đạt được độ chính xác cao như kì vọng (chỉ khoảng 77%), có thể lí do bắt nguồn từ việc lựa chọn đặc trưng kết cấu chưa phù hợp, do chưa đủ dữ liệu để huấn luyện thuật toán, hoặc do mô hình học tập vẫn chưa thật sự tốt. Vì vậy, hướng phát triển trong tương lai của nhóm chúng em là sẽ nghiên cứu và thử nghiệm trên nhiều dạng mô hình học máy hơn, cung cấp nhiều dữ liệu hơn cũng như tìm hiểu thêm về các đặc trưng kết cấu khác của Haralick.

TÀI LIỆU THAM KHẢO

- [1.] Hall-Beyer, Mryka. "GLCM texture: a tutorial." National Council on Geographic Information and Analysis Remote Sensing Core Curriculum 3.1 (2000): 75.
- [2.] Benco, Miroslav, et al. "An advanced approach to extraction of colour texture features based on GLCM." International Journal of Advanced Robotic Systems 11.7 (2014): 104.
- [3.] Mohanaiah, P., P. Sathyanarayana, and L. GuruKumar. "Image texture feature extraction using GLCM approach." International journal of scientific and research publications 3.5 (2013): 1-5.
- [4.] Alazawi, Sundos Abdulameer, Narjis Mezaal Shati, and Amel H. Abbas. "Texture features extraction based on GLCM for face retrieval system." Periodicals of Engineering and Natural Sciences 7.3 (2019): 1459-1467.
- [5.] Mutlag, Wamidh K., et al. "Feature extraction methods: a review." Journal of Physics: Conference Series. Vol. 1591. No. 1. IOP Publishing, 2020.
- [6.] "GLCM Texture Feature." Echoview, support.echoview.com/WebHelp/Windows_and_Dialog_Boxes/Dialog_Boxes/Variable_properties_dialog_box/Operator_pages/GLCM_Texture_Features.htm#Texture_page. Accessed 1 May 2022.
- [7.] CodeLearn. "Jupyter Notebook - Công Cụ Python Cơ Bản Cho Beginner." CodeLearn, codelearn.io/sharing/jupyter-notebook-tutorial. Accessed 1 May 2022.
- [8.] "Properties of Gray-Level Co-Occurrence Matrix - MATLAB Graycoprops." MathWorks, www.mathworks.com/help/images/ref/graycoprops.html. Accessed 29 April 2022.
- [9.] Bnsreenu. "GitHub - Bnsreenu/Python_for_microscopists: https://www.youtube.com/channel/UC34rWHtPJulxr5wp2Xa04w?Sub_confirmation=1." GitHub, github.com/bnsreenu/python_for_microscopists. Accessed 29 Apr. 2022.