

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN



BÁO CÁO MÔN HỌC
NHẬP MÔN THỊ GIÁC MÁY TÍNH – XỬ LÝ ẢNH
VÀ ỨNG DỤNG
ĐỀ TÀI: NHẬN DIỆN ĐỊA ĐIỂM DU LỊCH

Giảng viên hướng dẫn: TS. Mai Tiến Dũng

Lớp: CS231.M11 và CS406.M11

Sinh viên thực hiện:	MSSV:
Đỗ Minh Trí	19520307
Phan Trọng Hậu	19520077
Cao Đức Trí	19520305

Tp Hồ Chí Minh, tháng 12 năm 2021

This image shows a full page of white paper with horizontal dotted lines. The lines are evenly spaced and run across the width of the page, providing a guide for handwriting or typing. There are no margins, text, or other markings on the page.

Chữ kí giáo viên

MỤC LỤC

CHƯƠNG I: GIỚI THIỆU ĐỀ TÀI, MÔ TẢ DỮ LIỆU	1
1. Giới thiệu đề tài:.....	1
1.1 Tình huống bài toán:	1
1.2 Input và Output:	1
2. Cách thu thập bộ dữ liệu:	2
3. Mô tả dữ liệu:	5
4. Khó khăn của bộ dữ liệu:	8
CHƯƠNG II: TĂNG CƯỜNG DỮ LIỆU.....	10
1. Flip:	10
2. Color:	10
CHƯƠNG III: SỬ DỤNG CÁC MODEL ĐỂ ĐÁNH GIÁ.....	11
1. Sử dụng các hàm để trích xuất 3 đặc trưng của ảnh là màu sắc, hình dạng và kết cấu – hoa văn:.....	11
1.1. Color histogram (màu sắc) :.....	11
1.2. Hu Monents (hình dạng):	13
1.3. Haralick Texture (kết cấu – hoa văn):	14
2. Các loại tranfer learning của VGG16:	16
2.1. Feature Extraction của VGG16:	17
2.2. Fine-tuning của VGG16:.....	18
CHƯƠNG IV: KẾT QUẢ THỰC NGHIỆM, NHẬN XÉT VÀ DEMO:	20
1. Kết quả thực nghiệm và nhận xét:	20
2. Kết quả demo:	22
TÀI LIỆU THAM KHẢO	24

CHƯƠNG I: GIỚI THIỆU ĐỀ TÀI, MÔ TẢ DỮ LIỆU

1. Giới thiệu đề tài:

1.1 Tình huống bài toán:

Ngày nay, nhu cầu đi du lịch của giới trẻ đang ngày càng cao, sự tò mò và muốn khám phá xung quanh là điều tất yếu. Đặc biệt là sau khi đợt dịch Covid 19 kết thúc, khi mà trước đó mọi người đều phải hạn chế ra đường để đảm bảo an toàn thì nhu cầu tìm kiếm địa điểm du lịch để xả stress của mọi người là rất cao. Vấn đề đặt ra ở đây là khi chúng ta tình cờ nhìn thấy một bức ảnh chụp trên facebook, instagram..., một nơi nào đó rất đẹp mà ta muốn biết để lưu lại và có dịp thì sẽ đến. Tuy nhiên để đến được nơi trong bức ảnh thì điều đầu tiên chúng ta cần phải biết là tên của địa điểm có trong bức ảnh đó để có thể sử dụng google map tìm vị trí của nó trên bản đồ. Đó là lý do để nhóm tụi em thực hiện đề án này.

1.2 Input và Output:

- Input: một bức ảnh chụp cảnh quan của một địa điểm nào đó có thể lấy từ mạng xã hội hoặc là một bức ảnh tự chụp.



Hình 1.1.1



Hình 1.1.2

- Output: Tên của địa điểm du lịch đó mà bức ảnh input thể hiện

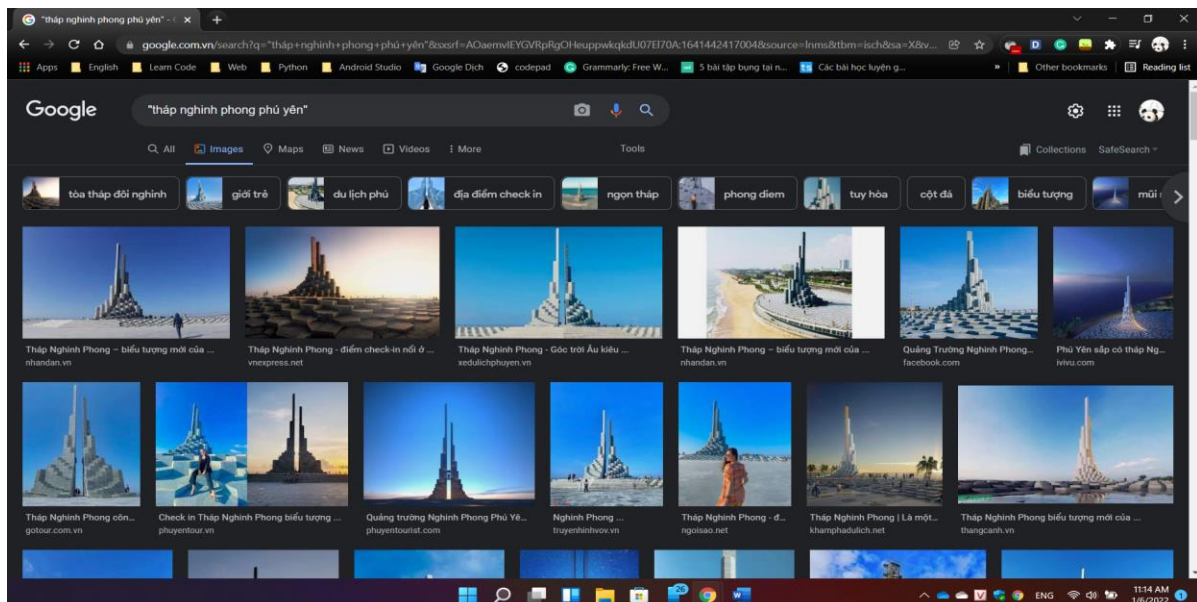


→ Tháp Poklong Gairai - Ninh Thuận

Hình 1.1.3

2. Cách thu thập bộ dữ liệu:

- Dữ liệu phù hợp với ngữ cảnh của bài toán này xuất phát từ ảnh trên mạng (google, facebook, instagram,...), hoặc là ảnh tự chụp. Vì ngữ cảnh của bài toán khá lớn, rất khó để nhóm có thể tự thu thập hết toàn bộ dữ liệu của tất cả các địa điểm du lịch trên toàn thế giới nên trong đề án này tụi em sẽ chỉ giới hạn các địa điểm du lịch ở ba tỉnh thành của Việt Nam là Phú Yên, Ninh Thuận và TP. Hồ Chí Minh.
- Thu thập thủ công bằng việc lấy ảnh trên mạng (facebook, google, instagram,...) theo từ khóa tìm kiếm.



Hình 1.2.1

- Để xác nhất với thực tế và training hiệu quả thì đòi hỏi bức ảnh input phải lấy được tổng quan đặc trưng phong cảnh với nhiều góc độ chụp khác nhau trong các hoàn cảnh và điều kiện thời tiết khác nhau, ảnh có thể là:

- Một bức ảnh toàn cảnh không người hoặc có người



Hình 1.2.2 (Ảnh không người)



Hình 1.2.3 (Ảnh có người)

- Ảnh selfie với phong cảnh.



Hình 1.2.4 Ảnh selfie với phong cảnh

- Và ảnh chụp có thể là buổi sáng hay tối tùy thuộc vào thời gian hoạt động du lịch của địa điểm.



Hình 1.2.5: Ảnh ban ngày



Hình 1.2.6: Ảnh ban đêm

- Ảnh có thể đã qua chỉnh sửa

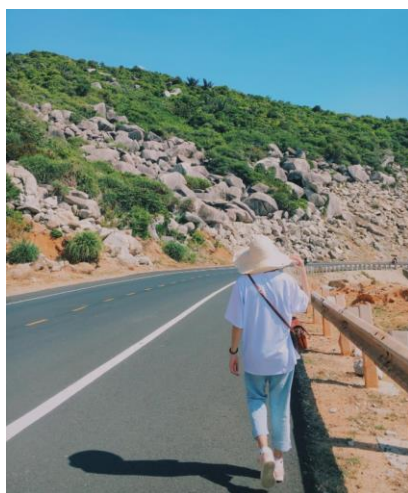


Hình 1.2.7



Hình 1.2.8

- Đối với những bức ảnh không có quá nhiều nét đặc trưng riêng biệt của địa điểm đó thì sẽ rất khó trong việc nhận dạng ví dụ như ảnh chụp đường đi, bầu trời,... của một địa điểm.



Hình 1.2.9

3. Mô tả dữ liệu:

- Số lượng ảnh: 2080 ảnh bao gồm 52 địa điểm thuộc 3 tỉnh thành (Tp Hồ Chí Minh, Ninh Thuận, Phú Yên) đồng nghĩa với 52 lớp.

- Tp Hồ Chí Minh: 776 ảnh với 19 địa điểm
- Ninh Thuận: 661 ảnh với 17 địa điểm
- Phú Yên: 643 ảnh với 16 địa điểm

- Các hình ảnh thu thập được bao gồm 3 nhóm địa điểm du lịch chính là:

- Du lịch biển, đảo, sông, suối... (Khoảng 13 lớp): là nhóm địa điểm khó nhận biết nhất trong 3 nhóm vì các địa điểm này thường không có các đặc trưng riêng biệt để phân biệt, dễ gây nhầm lẫn trong lúc nhận dạng.



Hình 1.3.1 Vịnh Vĩnh Hy – Ninh Thuận



Hình 1.3.2 Vịnh Vũng Rô – Phú Yên

- Du lịch đồi, núi, đồng bằng,... (15 lớp): Ở nhóm này các phong cảnh địa điểm khá đa dạng.



Hình 1.3.3



Hình 1.3.4

- Du lịch địa điểm có các kiến trúc (Khoảng 22 lớp): Ở nhóm này thường có những đặc trưng kiến trúc là nhóm địa điểm dễ nhận dạng nhất trong 3 nhóm du lịch.



Hình 1.3.5

- Thống kê số lượng ảnh của mỗi địa điểm:

Suối Lở Ổ - Ninh Thuận	50
Đường Sách - TP. Hồ Chí Minh	50
Land Mark 81 - Tp.Hồ Chí Minh	48
Hang Rải - Ninh Thuận	45
Đảo Bình Hưng - Ninh Thuận	45
Hồ điều hòa Hồ Sơn-Phú Yên	44
Tu viện Khánh An - Tp. Hồ Chí Minh	43
Hồ Con Rùa - Tp.Hồ Chí Minh	42
Dinh Độc Lập - Tp.Hồ Chí Minh	42
Núi Đá Bia-Phú Yên	42
Nhà thờ Tân Định - Tp. Hồ Chí Minh	42
Khu du lịch Suối Tiên - Tp. Hồ Chí Minh	42
Vườn nho Ba Mọi - Ninh Thuận	41
Ghềnh Đá Dĩa-Phú Yên	41
Bãi xếp gành Ông-Phú Yên	41
Chùa Thanh Lương-Phú Yên	41

Hình 1.3.6

Chợ Bến Thành - Tp.Hồ Chí Minh	41
Thác Chaper - Ninh Thuận	41
Bờ kè Khánh Hội - Ninh Thuận	41
Khu phố Nhật Bản- Tp.Hồ Chí Minh	41
Xóm Rớ-Phú Yên	41
Nhà thờ Mằng Lăng-Phú Yên	41
Núi Nhạn-Phú Yên	41
Bảo tàng Mỹ thuật - Tp.Hồ Chí Minh	41
Cầu Ông Cọp-Phú Yên	41
Bảo tàng lịch sử Việt Nam - Tp.Hồ Chí Minh	40
Đồi cát Nam Cương - Ninh Thuận	40
Long Vân garden-Phú Yên	40
Cầu ánh sao - Hồ Bán Nguyệt - Tp.Hồ Chí Minh	40
Cánh đồng hoa Thi Là - Ninh Thuận	40
Gành Đèn-Phú Yên	40
Vịnh Vũng Rô-Phú Yên	40
Bến nhà rồng - Tp.Hồ Chí Minh	40
Trùng Sơn Cổ Tự - Thiên viện trúc lâm Viên Ngộ - Ninh Thuận	40

Hình 1.3.7

Hải đăng Đại Lãnh(Mũi Điện)-Phú Yên	40
Tháp Nghinh Phong -Phú Yên	40
Cánh đồng rêu - Ninh Thuận	40
Bảo tàng tranh 3D Artinus - Tp.Hồ Chí Minh	40
Khu du lịch Tanyoli - Ninh Thuận	40
Chùa Bửu Long - Tp.Hồ Chí Minh	40
Hòn Yến-Phú Yên	40
Tháp PoKlong Garai - Ninh Thuận	40
Vịnh Vĩnh Hy - Ninh Thuận	40
Bưu điện - TP.Hồ Chí Minh	40
Khu du lịch văn hóa và sinh thái sen Charaih - Ninh Thuận	40
Tòa nhà Bitexco - Tp.Hồ Chí Minh	40
Nhà thờ Đức Bà - Tp.Hồ Chí Minh	40
Đập Đồng Cam-Phú Yên	30
Đồng cừu An Hòa - Ninh Thuận	29
Địa đạo Củ Chi - Tp.Hồ Chí Minh	25
Điện gió Dầm Nại - Ninh Thuận	25
Mũi Dinh - Ninh Thuận	24

Hình 1.3.8

4. Khó khăn của bộ dữ liệu:

- Do tình hình dịch Covid đang chuyển biến phức tạp nên nhóm đã không thể tự chụp ảnh mà chỉ có thể thu thập ảnh trên mạng nên hạn chế về số lượng và độ đa dạng của ảnh.
- Để nhận biết rõ địa điểm thì đòi hỏi địa điểm đó phải mang đặc trưng riêng biệt nếu không việc này sẽ làm khó khăn trong việc nhận dạng địa điểm đó. (Khoảng 7 địa điểm)



Hình 1.4.1

** Đảo Bình Hưng - Ninh Thuận không mang nhiều đặc trưng để phân biệt với một vùng biển bình thường khác*

- Có nhiều địa điểm tương đồng nhau gây sự nhầm lẫn trong việc nhận dạng (Khoảng 12 địa điểm).



Hình 1.4.2 Núi nhận - Phú Yên



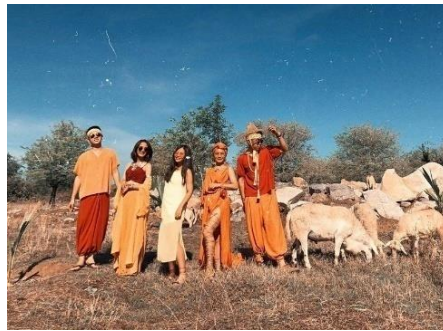
Hình 1.4.3 Tháp Poklong Gairai – Ninh Thuận

* 2 địa điểm có kiến trúc khá giống nhau dễ gây nhầm lẫn cho việc nhận dạng

- Các bức ảnh selfie phong cảnh hầu như bị người che mắt (Ít không đáng kể)



Hình 1.4.4



Hình 1.4.5

- Thời gian hoạt động của địa điểm cũng ảnh hưởng ít nhiều trong việc nhận diện địa điểm đó vì màu sắc chủ đạo của bức ảnh có thể chuyển từ sáng sang tối tùy ý rất dễ gây nhầm lẫn trong việc nhận diện.



Hình 1.4.6



Hình 1.4.7

- Cùng một địa điểm nhưng có thể chụp từ nhiều góc độ và vị trí khác nhau cũng sẽ gây ảnh hưởng đến quá trình nhận diện của địa điểm đó.



Hình 1.4.8



Hình 1.4.9

CHƯƠNG II: TĂNG CƯỜNG DỮ LIỆU

Với những khó khăn trong việc nhận diện địa điểm vừa nêu ở trên thì nhóm đã đề xuất 2 giải pháp tăng cường dữ liệu là flip và color để khắc phục một vài điểm khó khăn ấy.

1. Flip:

- Flip là 1 phép biến đổi hình ảnh nhờ vào phép đối xứng trục, hình ảnh được tạo ra có thể bị lật lên hoặc phản chiếu theo hướng ngược lại giống trong gương.



Hình 2.1.1



Hình 2.1.2

- Hình bên phải được tạo ra nhờ vào phép biến hình flip cho ra 1 bức ảnh có 1 góc nhìn khác.

2. Color:

- Color là cách ta tăng giảm độ sáng của bức ảnh để cho ra một bức ảnh tối hoặc sáng hơn so với ảnh gốc. Bởi vì phần lớn dữ liệu là ảnh ban ngày nên ta sẽ giảm độ sáng để bức ảnh tối lại và có cảm giác như có sự thay đổi về thời gian.



Hình 2.2.1



Hình 2.2.2

- Hình bên phải được tạo ra nhờ giảm độ sáng của 3 kênh màu (đỏ, xanh, vàng) làm như vậy ảnh có cảm giác bị tối đi.

CHƯƠNG III: SỬ DỤNG CÁC MODEL ĐỂ ĐÁNH GIÁ

Vì độ chính xác của model phụ thuộc vào cách trích xuất đặc trưng của dữ liệu nên nhóm sẽ thử sử dụng hai cách trích xuất đặc trưng của bộ dữ liệu như sau:

1. Sử dụng các hàm để trích xuất 3 đặc trưng của ảnh là màu sắc, hình dạng và kết cấu – hoa văn:

Bước đầu tiên của quá trình nhận dạng ảnh đó là xây dựng được các bộ mô tả hình ảnh (để trích xuất ra được đặc trưng quan trọng của dữ liệu). Chúng ta cần xác định rõ đặc điểm nào của các bức ảnh sẽ được sử dụng để phân biệt chúng, có thể là màu sắc chủ đạo, hình dạng của đối tượng trong bức ảnh, kết cấu - hoa văn hoặc là kết hợp các đặc điểm này lại. Ở đây chúng ta trích xuất feature một bức ảnh dựa trên cả ba đặc điểm đó.

1.1. Color histogram (màu sắc) :

- Color histogram là một dạng đặc trưng toàn cục biểu diễn phân phối của các màu trên ảnh. Color histogram thống kê số lượng các pixel có giá trị nằm trong một khoảng màu nhất định cho trước. Color histogram có thể tính trên các dạng ảnh RGB hoặc HSV, thông dụng là HSV (Hue – vùng màu, Saturation – độ bão

hòa màu, Value – độ sáng). Ví dụ như ở các bức ảnh du lịch biển thì màu sắc chủ đạo sẽ là xanh dương và thường là ảnh ban ngày, còn ở các bức ảnh du lịch kiến trúc màu chủ đạo sẽ có nhiều thay đổi khác nhau và có thể có ảnh ban đêm.[2]



Hình 3.1.1 Cầu Ánh Sao – Hồ Bán Nguyệt – TP.HCM



Hình 3.1.2. Đảo Bình Hưng – Ninh Thuận

- Trích xuất đặc trưng của 1 ảnh bằng color histogram :



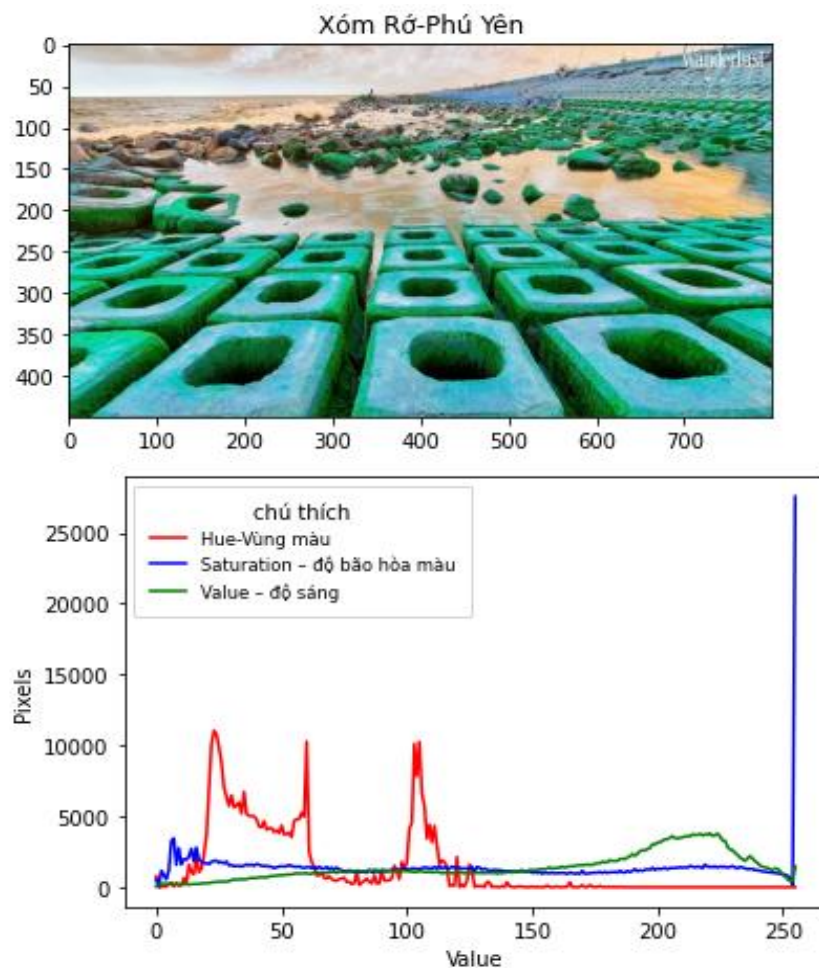
Hình 3.1.3

- Cài đặt trên Python:

```
# trích xuất đặc trưng theo : Color Histogram
def fd_histogram(image, mask=None):
    bins = 8
    # chuyển ảnh về dạng HSV color-space
    image = cv2.cvtColor(image, cv2.COLOR_BGR2HSV)
    # tính toán ảnh về dạng Color Histogram
    hist = cv2.calcHist([image], [0, 1, 2], None, [bins, bins, bins], [0, 256, 0, 256, 0, 256])
    # normalize về dạng histogram
    cv2.normalize(hist, hist)
    return hist.flatten()
```

Hình 3.1.4

- Đồ thị mô tả thống kê số lượng các pixel có giá trị nằm trong một khoảng màu nhất định cho trước trên dạng ảnh HSV:



Hình 3.1.5

1.2. Hu Moments (hình dạng):

- Hu Moments là một Image Descriptor sử dụng các phép thống kê để mô tả hình dạng của một đối tượng có trong bức ảnh nhị phân hoặc edged-image. Hu Moments Image Descriptor trả về một Feature Vector gồm 7 giá trị. Feature Vector này sẽ được so sánh với nhau để xác định sự tương đồng giữa hai vật thể. Ví dụ ở những địa điểm có các vật thể gần giống nhau thì ta sử dụng cách trích xuất này để phân biệt các hình ảnh.

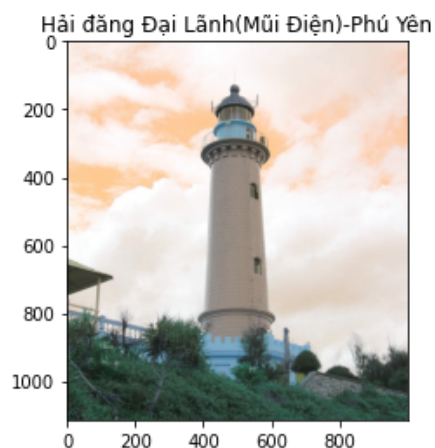


Hình 3.1.6 Hải đăng Đại Lãnh – Phú Yên



Hình 3.1.7 Hải đăng Mũi Dinh – Ninh Thuận

- Trích xuất đặc trưng của 1 ảnh bằng Hu Moments:



số chiều của vector của ảnh sau dùng Hu Moment : 7

vector sau khi trích xuất của ảnh : [8.18565307e-04 2.48629983e-09 3.97764772e-12 1.71688750e-12 2.37935763e-24 -3.41305911e-17 -3.80381793e-24]

Hình 3.1.8

- Cài đặt trên Python:

```
def fd_hu_moments(image):
    image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
    feature = cv2.HuMoments(cv2.moments(image)).flatten()
    return feature
```

Hình 3.1.9

1.3. Haralick Texture (kết cấu – hoa văn):

- Haralick được dùng để mô tả kết cấu (texture) và hoa văn (pattern) của một bức ảnh, đối tượng, bao gồm vẻ bề ngoài (appearance), sự nhất quán (consistency) và cảm giác về bề mặt (“feeling of surface”) có trong bức ảnh. Cách trích xuất này

được dùng để so sánh các bức ảnh có màu sắc và hình dạng vật thể gần như tương đồng nhưng kết cấu, bố cục của bức ảnh khác nhau.

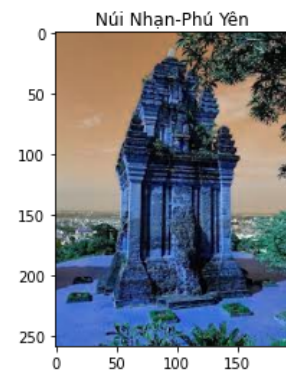


Hình 3.1.10 Xóm Rớ - Phú Yên



Hình 3.1.11 Cánh đồng rêu – Ninh Thuận

- Trích xuất đặc trưng của 1 ảnh bằng Haralick Texture:



số chiều của vector của ảnh sau dùng Haralick Texture : 13
vector sau khi trích xuất của ảnh : [6.66790637e-04 9.42113977e+02 8.15290657e-01 2.55012623e+03
2.41392079e-01 2.06782594e+02 9.25839093e+03 8.37480397e+00
1.29676587e+01 2.03471083e-04 5.42200482e+00 -2.56188478e-01
9.87970185e-01]

Hình 3.1.12

- Cài đặt trên Python:

```
# trích xuất đặc trưng theo : Haralick Texture
def fd_haralick(image):
    # chuyển ảnh về dạng grayscale
    gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
    # tính toán chuyển thành vector haralick texture feature
    haralick = mt.features.haralick(gray).mean(axis=0)
    # trả về kết quả
    return haralick
```

Hình 3.1.13

- Sau đó , tổng hợp tất cả các cách trích xuất trên lại với nhau , tạo thành bộ các feature mới .

```
def preTrainImage_1(image):

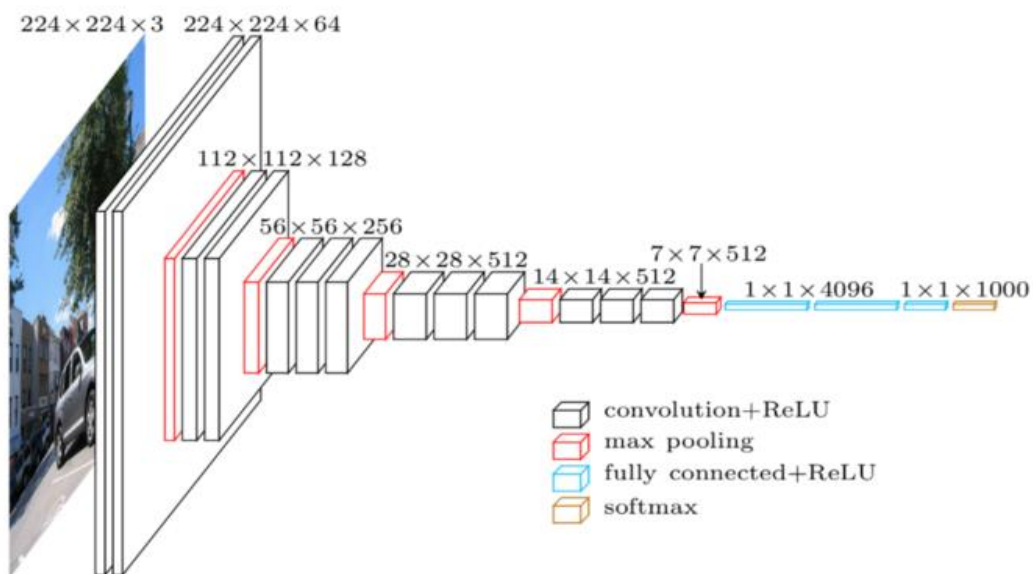
    fv_hu_moments = fd_hu_moments(image)
    fv_haralick    = fd_haralick(image)
    fv_histogram   = fd_histogram(image)
    # kết hợp các cách trích xuất
    feature = np.hstack([fv_histogram, fv_hu_moments, fv_haralick])
    return feature
```

Hình 3.1.14

- Số chiều của 1 ảnh sau khi trích xuất các đặc trưng của bộ dữ liệu ban đầu dựa 3 đặc trưng của ảnh là 532.

2. Các loại tranfer learning của VGG16:

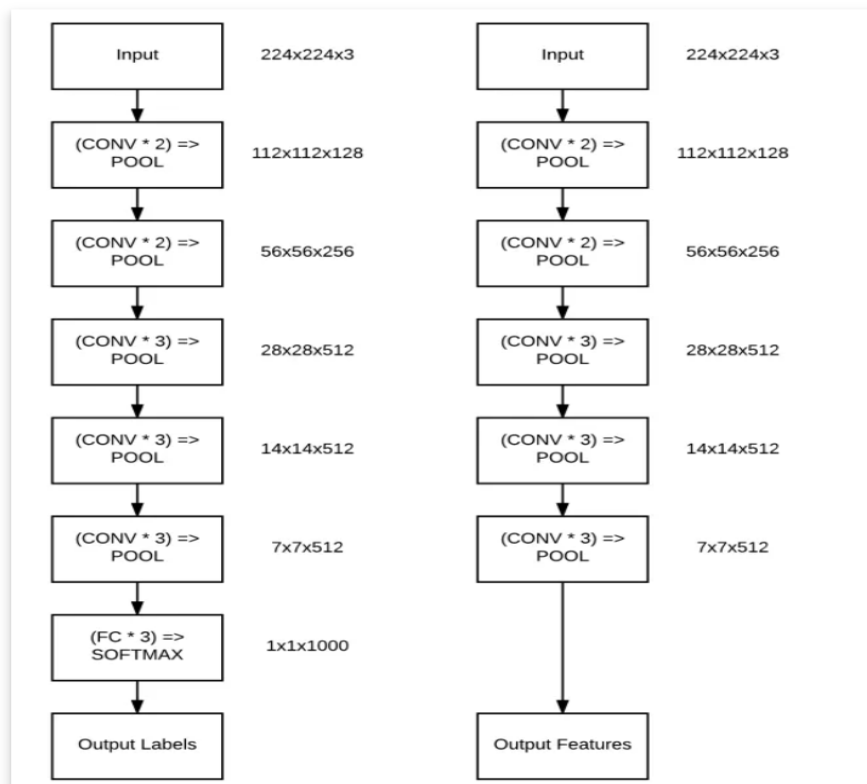
Mạng VGG16 được đề xuất bởi K.Simonyan và A.Zisserman của đại học Oxford, áp dụng cho các bài toán phân loại ảnh trên tập dataset ImageNet 1000 lớp, thông thường người ta hay sử dụng VGG16 để rút trích đặc trưng học sâu cho một bức ảnh. Như vậy input là một bức ảnh và output là một dãy số có 4096 giá trị được xem như là vecto đặc trưng cho ảnh đó.



Hình 3.2.1

2.1. Feature Extraction của VGG16:

Là quá trình chuyển đổi tập dữ liệu thô ban đầu thành các tập thuộc tính có thể giúp biểu diễn tập dữ liệu ban đầu tốt hơn, tạo điều kiện để giải quyết các bài toán dễ dàng hơn, giúp tương thích với từng mô hình dự đoán cụ thể cũng như cải thiện độ chính xác của mô hình dự đoán hiện tại.

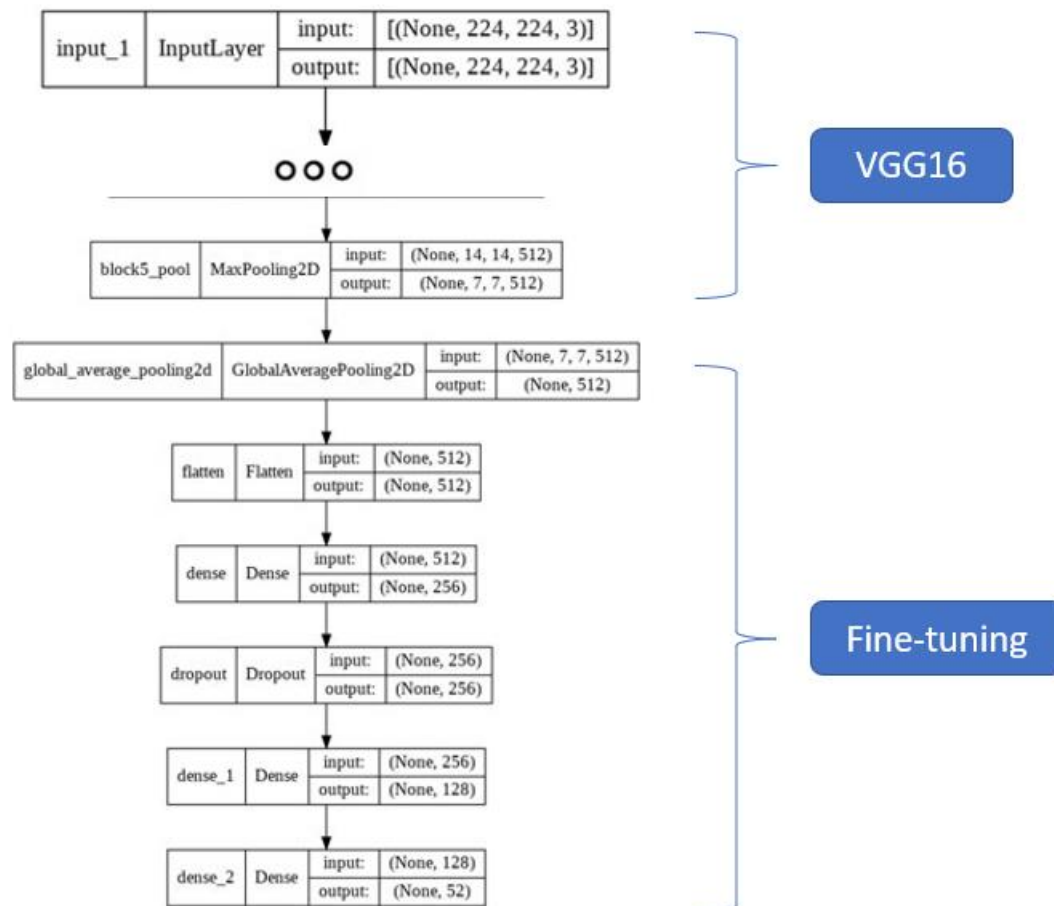


Bên trái là mô hình VGG16, bên phải là mô hình VGG16 chỉ bao gồm ConvNet (bỏ Fully connected layer)

Hình 3.2.2

2.2. Fine-tuning của VGG16:

a. Mô tả cấu trúc Fine-tuning của VGG16:

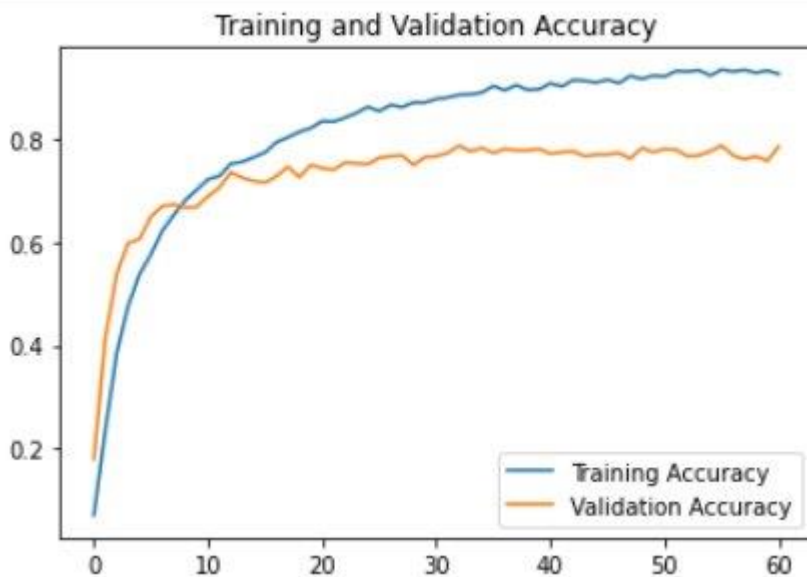


Hình 3.2.3: Fine-tuning của VGG16

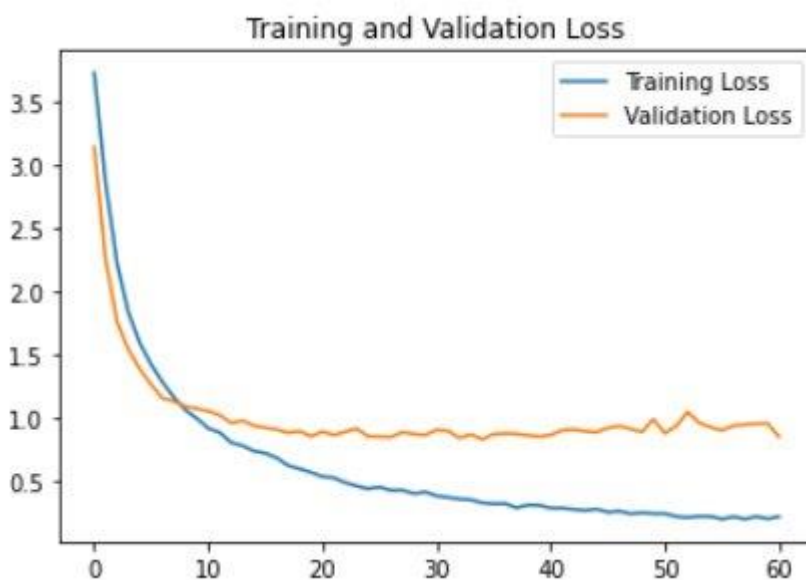
- Input Layer nhận ảnh có kích thước $224 \times 224 \times 3$.
- Ta sử dụng cấu trúc của VGG16 đến lớp Max Pooling cuối cùng, sau đó ta bắt đầu fine-tuning bằng các lớp sau đây:
 - + Lớp Global Average Pooling 2D.
 - + Lớp Flatten.
 - + Lớp Dense với 512 unit kết hợp với hàm Relu.
 - + Lớp Dropout với tỷ lệ là 0.5, giúp giảm hiện tượng overfitting của mô hình.
 - + Lớp Dense với 256 unit kết hợp với hàm Relu
 - + Lớp Dense cuối cùng với 52 unit tương ứng với 52 địa điểm của bộ dữ liệu và kết hợp với hàm Softmax được sử dụng rộng rãi cho bài toán Phân loại nhiều lớp (Multi-Class Classification).
- b. Các thông số được dùng trong mô hình Fine-tuning của VGG16:

- + Số Epochs : 100.
- + Hàm Loss : Categorical crossentropy được sử dụng rộng rãi cho bài toán Phân loại nhiều lớp (Multi-Class Classification).
- + Thuật toán optimizer sử dụng : Adam với learning rate là 0.001.
- + Kích thước batch size là 64.

c. Kết quả của quá trình huấn luyện mô hình Fine-tuning:



Hình 3.2.4: Biểu đồ mô tả accuracy của quá trình training và validation



Hình 3.2.5: Biểu đồ mô tả điểm loss của quá trình training và validation

d. Đánh giá:

Mô hình Fine-tuning càng về các epoch lớn hơn 50 càng gặp vấn đề overfitting và vì sử dụng hàm EarlyStopping nên dừng lại sớm ở số epoch số 61.

CHƯƠNG IV: KẾT QUẢ THỰC NGHIỆM, NHẬN XÉT VÀ DEMO:

1. Kết quả thực nghiệm và nhận xét:

- Nhóm em sử dụng 70 % ảnh từ bộ dữ liệu (1440 ảnh) dùng để train và 30 % ảnh (640 ảnh) dùng để test.
- Dùng 1440 ảnh để tăng cường dữ liệu thu được 4320 ảnh dùng để train

Tên model	Loss	Accuracy
Trích xuất các đặc điểm + Logistic Regression	2.6048872305778437	0.3421875
Features Extraction using VGG16 + Logistic Regression	0.479148473228223	0.871875
Fine-tuning using VGG16	0.8509	0.7859

Hình 4.1: Kết quả thu được từ các mô hình

- Dựa vào hình trên, mô hình sử dụng phương pháp Features Extraction của VGG16 kết hợp với thuật toán Logistic Regression cho accuracy lớn nhất là 0.87 và giá trị loss thấp nhất với 0.479.
- Vì vậy, nhóm em sử dụng mô hình này để tính các giá trị precision, recall, f1-score.

	precision	recall	f1-score	support
Bãi xếp gành Ông-Phú Yên	0.722222	1	0.83871	13
Bưu điện - TP.Hồ Chí Minh	1	0.666667	0.8	12
Bảo tàng Mỹ thuật - Tp.Hồ Chí Minh	0.764706	1	0.866667	13
Bảo tàng lịch sử Việt Nam - Tp.Hồ Chí Minh	0.923077	1	0.96	12
Bảo tàng tranh 3D Artinus - Tp.Hồ Chí Minh	0.923077	1	0.96	12
Bến nhà rồng - Tp.Hồ Chí Minh	0.846154	0.916667	0.88	12
Bờ kè Khánh Hội - Ninh Thuận	0.818182	0.692308	0.75	13
Chùa Bửu Long - Tp.Hồ Chí Minh	0.846154	0.916667	0.88	12
Chùa Thanh Lương-Phú Yên	0.857143	0.923077	0.888889	13
Chợ Bến Thành - Tp.Hồ Chí Minh	1	1	1	13
Cánh đồng hoa Thì Là - Ninh Thuận	0.8	1	0.888889	12
Cánh đồng rêu - Ninh Thuận	0.9	0.75	0.818182	12
Cầu Ông Cọp-Phú Yên	0.8	0.923077	0.857143	13
Cầu ánh sao - Hồ Bán Nguyệt - Tp.Hồ Chí Minh	0.857143	1	0.923077	12

Dinh Độc Lập - Tp.Hồ Chí Minh	1	0.923077	0.96	13
Ghềnh Đá Dĩa-Phú Yên	0.857143	0.923077	0.888889	13
Gành Đèn-Phú Yên	1	0.666667	0.8	12
Hang Rái - Ninh Thuận	0.8125	0.928571	0.866667	14
Hòn Yến-Phú Yên	1	1	1	12
Hải đăng Đại Lãnh(Mũi Điện)-Phú Yên	0.6	0.75	0.666667	12
Hồ Con Rùa - Tp.Hồ Chí Minh	1	0.923077	0.96	13
Hồ điều hòa Hồ Sơn-Phú Yên	0.933333	1	0.965517	14
Khu du lịch Suối Tiên - Tp. Hồ Chí Minh	0.909091	0.769231	0.833333	13
Khu du lịch Tanyoli - Ninh Thuận	1	0.666667	0.8	12
Khu du lịch văn hóa và sinh thái sen Charaih - Ninh Thuận	1	0.833333	0.909091	12
Khu phố Nhật Bản- Tp.Hồ Chí Minh	0.888889	0.615385	0.727273	13
Land Mark 81 - Tp.Hồ Chí Minh	0.933333	0.933333	0.933333	15
Long Vân garden-Phú Yên	0.818182	0.75	0.782609	12
Mũi Dinh - Ninh Thuận	1	0.625	0.769231	8
Nhà thờ Mằng Lăng-Phú Yên	1	1	1	13
Nhà thờ Tân Định - Tp. Hồ Chí Minh	1	0.846154	0.916667	13
Nhà thờ Đức Bà - Tp.Hồ Chí Minh	1	1	1	12
Núi Nhạn-Phú Yên	0.916667	0.846154	0.88	13
Núi Đá Bia-Phú Yên	0.8125	1	0.896552	13
Suối Lồ Ô - Ninh Thuận	0.764706	0.866667	0.8125	15
Thác Chaper - Ninh Thuận	1	0.923077	0.96	13
Tháp Nghinh Phong -Phú Yên	1	1	1	12
Tháp PoKlong Garai - Ninh Thuận	1	0.916667	0.956522	12
Trùng Sơn Cổ Tự - Thiền viện trúc lâm Viên Ngộ - Ninh Thuận	0.714286	0.833333	0.769231	12
Tu viện Khánh An - Tp. Hồ Chí Minh	1	0.846154	0.916667	13
Tòa nhà Bitexco - Tp.Hồ Chí Minh	0.846154	0.916667	0.88	12
Vườn nho Ba Mọi - Ninh Thuận	0.928571	1	0.962963	13
Vịnh Vĩnh Hy - Ninh Thuận	0.888889	0.666667	0.761905	12
Vịnh Vũng Rô-Phú Yên	0.692308	0.75	0.72	12
Xóm Rớ-Phú Yên	1	0.923077	0.96	13
Điện gió Dầm Nại - Ninh Thuận	0.875	0.875	0.875	8
Đường Sách - TP. Hồ Chí Minh	0.7	0.933333	0.8	15
Đảo Bình Hưng - Ninh Thuận	0.75	0.428571	0.545455	14
Đập Đồng Cam-Phú Yên	0.875	0.777778	0.823529	9
Địa đạo Củ Chi - Tp.Hồ Chí Minh	0.777778	0.875	0.823529	8
Đồi cát Nam Cương - Ninh Thuận	0.8	1	0.888889	12
Đồng cừu An Hòa - Ninh Thuận	0.8	0.888889	0.842105	9
accuracy	0.871875	0.871875	0.871875	0.871875
macro avg	0.883696	0.869405	0.868571	640
weighted avg	0.883205	0.871875	0.869651	640

Bảng 1: Classification report

- Dựa vào bảng trên, cho thấy các địa điểm như Tháp Nghinh Phong -Phú Yên, Nhà thờ Đức Bà - Tp.Hồ Chí Minh,... là những địa điểm mà mô hình này dự đoán chính xác nhất trên tập test.

2. Kết quả demo:

Source code: <https://github.com/hauphanlvc/DemoDoAnCS406>



Một số ảnh nhận diện đúng :



Features Extraction using VGG16 + Logistic Regression: Nhà thờ Măng Lăng-Phú Yên
 Fine-tuning using VGG16: Bưu điện - TP.Hồ Chí Minh
 Trích xuất các đặc điểm + Logistic Regression: Land Mark 81 - Tp.Hồ Chí Minh
 Chính xác: Nhà thờ Măng Lăng-Phú Yên

29



Một số ảnh nhận diện đúng :



Features Extraction using VGG16 + Logistic Regression: Chùa Bửu Long - Tp.Hồ Chí Minh
 Fine-tuning using VGG16: Nhà thờ Đức Bà - Tp.Hồ Chí Minh
 Trích xuất các đặc điểm + Logistic Regression: Land Mark 81 - Tp.Hồ Chí Minh
 Chính xác: Chùa Bửu Long - Tp.Hồ Chí Minh

30

Một số ảnh nhận diện sai :



Features Extraction using VGG16 + Logistic Regression: Núi Nhạn-Phú Yên
Fine-tuning using VGG16: Núi Nhạn-Phú Yên
Trích xuất các đặc điểm + Logistic Regression: Land Mark 81 - Tp.Hồ Chí Minh
Chính xác: Tháp PoKlong Garai - Ninh Thuận

31

Một số ảnh nhận diện sai :



Features Extraction using VGG16 + Logistic Regression: Hải đăng Đại Lãnh(Mũi Điện)-Phú Yên
Fine-tuning using VGG16: Hải đăng Đại Lãnh(Mũi Điện)-Phú Yên
Trích xuất các đặc điểm + Logistic Regression: Land Mark 81 - Tp.Hồ Chí Minh
Chính xác: Gành Đèn-Phú Yên

32

TÀI LIỆU THAM KHẢO

- [1] <https://cvexplained.wordpress.com/2020/07/21/10-4-hu-moments/>
- [2] <https://datacarpentry.org/image-processing/05-creating-histograms/>
- [3] <https://www.youtube.com/watch?v=DA4-JRpqXAU>
- [4] <https://towardsdatascience.com/step-by-step-vgg16-implementation-in-keras-for-beginners-a833c686ae6c>
- [5] <https://keras.io/api/applications/#vgg16>
- [6] <https://www.kaggle.com/tolgahancepel/feature-extraction-and-fine-tuning-using-vgg16>
- [7] <https://careerfoundry.com/en/blog/data-analytics/what-is-random-forest/>
- [8] <https://nttuan8.com/bai-6-convolutional-neural-network/>
- [9] <https://forum.machinelearningcoban.com/t/kien-truc-cac-mang-cnn-noi-tieng-phan-1-alex-lenet-inception-vgg/2582>
- [10] <https://viblo.asia/p/logistic-regression-bai-toan-co-ban-trong-machine-learning-924IJ4rzKPM>