

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



ĐỒ ÁN MÔN HỌC

Máy học

Học kỳ II (2019-2020)

NHẬN DIỆN CÁC TOÀ NHÀ TRONG
KHUÔN VIÊN TRƯỜNG
ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

Sinh viên 1: Nguyễn Khánh Toàn

MSSV: 18521509

Giảng viên: Lê Đình Duy

Phạm Nguyễn Trường An

Thành phố Hồ Chí Minh, tháng 8 năm 2020

Phần I. PHẦN MỞ ĐẦU

1. Giới thiệu đề tài

- Bài toán phân lớp (Classification) là một trong những bài toán lớn trong lĩnh vực máy học (Machine Learning), vì thế ở đề án cuối kì cho môn học Máy học, em lựa chọn bài toán “Nhận diện các toà nhà trong khuôn viên trường Đại học Công nghệ thông tin”.

Với bài toán này, dữ liệu được thu thập bằng việc chụp ảnh các toà nhà sau đó sử dụng các thuật toán trong máy học cho bài toán phân lớp để xử lý.

- Bài toán này nếu được phát triển sẽ giúp đỡ những sinh viên mới, phụ huynh đến trường có thể dễ dàng hơn trong việc tìm các toà nhà, tìm đường khi lần đầu tới trường đại học Công nghệ thông tin.

2. Mục tiêu và quy trình tìm hiểu

– Mục tiêu:

- Nắm được quy trình giải một bài toán máy học
- Hiểu và vận dụng được các thuật toán trong máy học
- Sử dụng thành thạo các thư viện hỗ trợ
- Đánh giá bài toán và tìm hiểu nguyên nhân cho ra kết quả bài toán
- Giải quyết được bài toán và xây dựng hướng đi để đưa bài toán vào ứng dụng thực tế

– Quy trình:

- Thu thập dữ liệu (sử dụng camera sau điện thoại để chụp ảnh, khoảng 1100 tấm cho tập train và 400 tấm cho tập test)
- Tiền xử lý dữ liệu: đưa về ảnh xám kích thước 448x448
- Rút trích đặc trưng: sử dụng tích chập để detect các cạnh của toà nhà, sau đó dùng histogram of oriented gradients (HOG) để trích xuất đặc trưng
- Chọn mô hình: bài toán sử dụng 3 mô hình để huấn luyện
 - Support Vector Classification
 - Random Forest Classifier
 - Gaussian Naïve Bayes
- Huấn luyện: sử dụng Google colab, thư viện hỗ trợ: sklearn

- Tinh chỉnh thông số: sau khi thực hiện việc huấn luyện, thay đổi thông số ở các bước tiền xử lý dữ liệu rút trích đặc trưng để so sánh và chọn ra mô hình tốt nhất
 - Đánh giá: đánh giá kết quả đạt được và đưa ra lý do về hiệu quả của model, từ đó đưa ra hướng phát triển cho bài toán
- Mô tả:
- Đầu vào: ảnh của một toà nhà trong trường đại học Công nghệ thông tin
 - Đầu ra: tên của toà nhà đó.

Phần II. NỘI DUNG

1. Thu thập dữ liệu

Mô tả bộ dữ liệu:

- Số lượng: tập train: 1100 ảnh, tập test 400 ảnh
- Số lớp 5 lớp: toà B, C, D, E, căn tin.
- Thiết bị: camera sau điện thoại xiaomi mi A3, Iphone 7 plus
- Ảnh được chụp vào thời điểm, thời tiết khác nhau

Do cây cối nhiều nên bộ dữ liệu có phần bị nhiễu, và bộ dữ liệu còn ít (mỗi lớp có khoảng 220 ảnh)

Toà nhà	Tập Train	Tập Test
B	277	101
C	235	85
D	157	95
E	257	91
Căn tin	170	89
Tổng	1096	461

2. Tiền xử lý dữ liệu

- Ảnh ban đầu: tỉ lệ 1:1
- Ảnh chuyển sang ảnh xám và chuyển về kích thước 448*448

3. Trích xuất đặc trưng

* Sử dụng tích chập để detect các cạnh trong ảnh

Tích chập là phép toán tuyến tính sử dụng bộ lọc cho một ảnh.

Công thức tích chập giữa hàm ảnh $f(x, y)$ và bộ lọc $k(x, y)$ (kích thước $m \times n$):

$$k(x, y) \star f(x, y) = \sum_{u=-m/2}^{m/2} \sum_{v=-n/2}^{n/2} k(u, v) f(x - u, y - v)$$



* Sử dụng HOG features:

HOG (histogram of oriented gradients) là một feature descriptor được sử dụng trong computer vision và xử lý hình ảnh, dùng để detect một đối tượng. Các bước thực hiện HOG features extraction:

- Chuẩn hóa hình ảnh trước khi xử lý
- Tính toán gradient theo cả hướng x và y .
- Lấy phiếu bầu cùng trọng số trong các cell
- Chuẩn hóa các block
- Thu thập tất cả các biểu đồ cường độ gradient định hướng để tạo ra feature vector cuối cùng.

Với ảnh 448*448, thông số Hog features: orientations=8, pixels per cell = 16*16, cells per block = 4*4, sẽ cho ra

$$(448/16-1)*(448/16-1)*(4*4)*8 = 93312 \text{ features cho mỗi hình}$$

4. Chọn model

– Support vector classification:

- Với support vector machine (máy vector hỗ trợ) xây dựng một siêu phẳng sử dụng cho bài toán phân lớp nhị phân. Các điểm thuộc một class nằm về cùng một phía của siêu mặt phẳng đó và ngược phía với toàn bộ các điểm thuộc class còn lại.
- Mở rộng SVM cho bài toán phân lớp đa lớp thông qua việc sử dụng nhiều binary classifiers và các kỹ thuật như one-vs-one hoặc one-vs-rest.
- Model này được sử dụng thông qua thư viện Sklearn: `sklearn.svm.SVC`

– Random Forest Classifier: đây là phương pháp xây dựng một tập hợp rất nhiều cây quyết định và sử dụng phương pháp voting để đưa ra quyết định về biến target cần được dự báo. Thông qua việc sử dụng nhiều cây quyết định sẽ xác định được features nào thực sự quan trọng, features nào ít quan trọng.

- `Sklearn.ensemble.RandomForestClassifier`

– Gaussian Naïve Bayes: Naive Bayes là một thuật toán phân lớp được mô hình hoá dựa trên định lý Bayes trong xác suất thống kê:

$$P(y|X) = \frac{P(X|y)P(y)}{P(X)}$$

- $P(y|X)$ gọi là posterior probability: xác suất của mục tiêu y với điều kiện có đặc trưng X
- $P(X|y)$ gọi là likelihood: xác suất của đặc trưng X khi đã biết mục tiêu y
- $P(y)$ gọi là prior probability của mục tiêu y
- $P(X)$ gọi là prior probability của đặc trưng X

- Trong mô hình Naive Bayes, có hai giả thiết được đặt ra:

- Các đặc trưng đưa vào mô hình là độc lập với nhau. Tức là sự thay đổi giá trị của một đặc trưng không ảnh hưởng đến các đặc trưng còn lại.

- Các đặc trưng đưa vào mô hình có ảnh hưởng ngang nhau đối với đầu ra mục tiêu.

Khi đó, kết quả mục tiêu y để $P(y|X)$ đạt cực đại trở thành:

$$y = \operatorname{argmax}_y P(y) \prod_{i=1}^n P(x_i|y)$$

Chính vì hai giả thiết gần như không tồn tại trong thực tế trên, mô hình này mới được gọi là naive (ngây thơ). Tuy nhiên, chính sự đơn giản của nó với việc dự đoán rất nhanh kết quả đầu ra khiến nó được sử dụng rất nhiều trong thực tế trên những bộ dữ liệu lớn, đem lại kết quả khả quan.

Trong thư viện sklearn, GNB được gọi: `sklearn.naive_bayes.GaussianNB`

5. Huấn luyện

- Sử dụng google colab để huấn luyện sau khi đã thực hiện tiền xử lý dữ liệu và rút trích đặc trưng.
- Thời gian huấn luyện và test cho mỗi lần khoảng 30 phút cho 3 models.

Độ chính xác	Support Vector classification	Random Forest Classifier	Gaussian Naïve Bayes
Train	97.54%	100%	83.56%
Test	56.83%	50.98%	57.48%

6. Tinh chỉnh thông số

- Thực hiện việc thay đổi các thông số đối với trích xuất đặc trưng của HOG Features và thực hiện lại việc huấn luyện
- Thay đổi orientation, pixels per cell, cells per block..

7. Đánh giá các mô hình

7.1. Đánh giá

- Xảy ra hiện tượng overfit (quá khớp), đạt kết quả cao ở tập huấn luyện nhưng đạt kết quả thấp ở tập test.
- Mô hình cho tính tổng quát cao nhất: Gaussian Naïve Bayes
- Mô hình bị overfit nhiều nhất: Random Forest Classifier

7.2. Giải thích

- Random Forest Classifier là thuật toán mạnh với variance lớn (rất nhiều cây quyết định trong một ‘rừng’ cây) cho nên sẽ đạt kết quả cao so với tập train, với dữ liệu train quá nhỏ dẫn tới tình trạng overfit.
- Với Gaussian Naïve Bayes tính toán dựa trên phân phối, hình dạng chính là đặc trưng của mỗi toà nhà, vì thế thuật toán này giải quyết tốt hơn nhờ tính toán dựa trên HOG Features (detect vật thể trong hình)
- Bộ dữ liệu còn nhỏ và chứ nhiều nhiễu (do cây cối trong trường quá nhiều)
- Một số góc chụp ở toà B và E có phần giống nhau dẫn tới kết quả thấp

Phần III. TỔNG KẾT

1. Đánh giá

- Đã xác định rõ bài toán.
- Thực hiện đủ 7 bước giải bài toán máy học.
- Thực hiện việc tìm hiểu các phương pháp, tài liệu mới.
- Tham khảo các dạng bài toán tương tự từ nhiều nguồn khác nhau.
- Vận dụng được các kiến thức trong môn máy học để hoàn thành đồ án.

2. Hướng phát triển.

- Để tăng độ hiệu quả cần thu thập thêm nhiều dữ liệu hơn.
- Cần cải thiện bước rút trích đặc trưng
- Thực hiện trên nhiều mô hình hơn nữa, có thể sử dụng các thuật toán học sâu để giải quyết.
- Ứng dụng thực tiễn: bài toán có thể dựa vào hình ảnh từ đó giúp cho người sử dụng biết họ đang ở đâu, mở rộng ra có thể trong các khu phố đông đúc dễ bị lạc (trong trường hợp không có internet, bản đồ...)