

Nhận dạng ảnh khung cảnh sử dụng bag of words

Chuẩn bị:

- Download source code và dữ liệu tại: <http://cs.brown.edu/courses/csci1430/2013/proj3/proj3.zip>
- Download thư viện [VLFeat 0.9.20 binary package](#)
- Tham khảo VL Feat Matlab reference: <http://www.vlfeat.org/matlab/matlab.html>

Giới thiệu:

Mục tiêu của bài tập lớn này là để các học viên làm quen với nhận dạng ảnh sử dụng đặc trưng ảnh và huấn luyện bộ phân lớp. Đầu tiên chúng ta sẽ bắt đầu với một phương pháp đơn giản là sử dụng đặc trưng ảnh nhỏ thu nhỏ (tiny image) và bộ phân loại nearest neighbor, sau đó sẽ sử dụng đặc trưng bag of word kết hợp với bộ phân loại SVM để xem kết quả được cải tiến như thế nào.

Trong bài tập này học viên sẽ phân loại các ảnh cảnh vật vào 1 trong 15 loại cảnh vật có trong tập dữ liệu (office, kitchen, living room,...).



Chi tiết:

Toàn bộ source code ở trong thư mục proj3. Chạy file proj3.m để thực hiện nhận dạng ảnh nhưng nếu không thay đổi code thì nó sẽ chọn random các cảnh vào các lớp và độ chính xác khoảng 7%.

Yêu cầu của bài tập này là phải cài đặt 2 phương pháp biểu diễn ảnh là tiny images và bag of SIFT và 2 phương pháp phân loại ảnh là nearest neighbor và linear SVM. (Ở trong bài tập này SIFT được sử dụng như một hàm, hiểu là một loại đặc trưng được

trích chọn để biểu diễn ảnh, kết quả sẽ cho ra một vector biểu diễn. Chi tiết về SIFT học viên sẽ được học chi tiết ở buổi học sau).

Yêu cầu báo cáo kết quả của các cài đặt sau:

- Tiny images representation and nearest neighbor classifier (accuracy of about 18-25%).
- Bag of SIFT representation and nearest neighbor classifier (accuracy of about 50-60%).
- Bag of SIFT representation and linear SVM classifier (accuracy of about 60-70%).

Học viên nên cài đặt theo thứ tự như trên và bắt đầu bằng việc cài đặt phương pháp đơn giản là Tiny images representation and nearest neighbor classifier để chạy thử.

Tiny image là một trong những loại biểu diễn đặc trưng ảnh một cách đơn giản nhất. Phương pháp này resize kích thước ảnh thành ảnh có độ phân giải nhỏ (nên dùng 16x16). Biểu diễn này thường không tốt vì nó bỏ qua những nội dung ở vùng tần số cao của ảnh và nó không bất biến với các phép biến đổi ảnh. Cài đặt tiny image trong file `get_tiny_images.m`

Nearest neighbor classifier là một phương pháp phân loại khá đơn giản. Khi ta cần phân loại 1 mẫu dữ liệu của tập test, thì ta tìm dữ liệu ở tập train gần nó nhất và gán nhãn giống nhãn ở tập train cho nó (ở đây sử dụng khoảng cách euclid). Tham khảo thêm và cài đặt phương pháp này tại file `nearest_neighbor_classify.m`

Sau khi đã thử nghiệm với Tiny image và nearest neighbor thì ta sẽ chuyển sang sử dụng bag of features (ở đây là bag of SIFT) thay cho tiny image. Trước khi biểu diễn tập dữ liệu bằng bag of SIFT histogram thì ta cần khởi tạo bộ từ điển. Tạo bộ từ điển bằng cách phân cụm (cluster) các đặc trưng SIFT. Tham khảo và cài đặt thêm ở file `get_bags_of_sifts.m`.

Có nhiều cách cài đặt tham số của bag of SIFT (number of clusters, sampling density, sampling scales, SIFT parameters, etc.). Điều chỉnh tham số và chạy thử SIFT và nearest neighbor Độ chính xác khoảng 50 - 60%.

Bước cuối cùng ta sẽ dùng 1-vs-all SVM là một phương pháp rất phổ biến để phân lớp. Phương pháp này hiệu quả hơn nearest neighbor. Kết quả của nearest neighbor không tận dụng được những thông tin của nhiều ảnh hợp lại mà chỉ dựa vào ảnh gần nó nhất. SVM nguyên bản chỉ phân biệt được 2 lớp, mà bài toán này có 15 lớp riêng biệt. Vì thế ta sẽ train tập dữ liệu theo phương pháp 1-vs-all, nghĩa là ta sẽ train tập dữ liệu để phân biệt dữ liệu có thuộc một tập hay không thuộc tập đó, ví dụ: 1 classifier sẽ phân loại tập kitchen và không phải kitchen, office và không phải office,... Vì thế ta sẽ có 15

classifier. Khi test thì tất cả các classifier được đưa vào để test trên mỗi ảnh test và classifier nào có điểm số trả về cao nhất thì sẽ chiến thắng. Ví dụ: bộ phân loại kitchen trả về điểm là -0.2 (trong đó 0 là biên) bộ office trả về điểm là -0.3 và các bộ khác có điểm thấp hơn thì ta sẽ chọn ảnh này thuộc lớp kitchen (mặc dù điểm trả về là âm nhưng nó vẫn được phân vào lớp kitchen). Trong SVM có tham số lambda ta phải thử nghiệm lambda để có kết quả tốt.

Chi tiết và cài đặt thêm ở svm_classify.m

Các file cần sửa cài đặt là: get_tiny_images.m, nearest_neighbor_classify.m, build_vocabulary.m, get_bags_of_sifts.m, and svm_classify.m
Create_results_webpage.m là file để tạo kết quả

Đánh giá:

Đánh giá dựa vào độ chính xác và bảng kết quả như mẫu: [table of true positives, false positives, and false negatives as a webpage](#)

Dữ liệu:

Tập dữ liệu có 100 ảnh train và test cho mỗi lớp.

Các hàm quan trọng cần dùng:

Lưu ý: Thư viện VLFeat sử dụng vectors cột nên phải chuyển vector hàng về cột để sử dụng được vlfeat:

[vl_dsift\(\)](#). Hàm này trả về SIFT descriptors

[vl_kmeans\(\)](#). Hàm này thực hiện thuật toán k-mean có thể sử dụng khi tạo từ điển.

[vl_svmtrain\(\)](#). Hàm này trả về các tham số của linear decision boundary (a hyperplane in your feature space). Học viên có thể sử dụng khoảng cách từ hyperplane để tính điểm của các bộ phân lại 1-vs-all.

[vl_alldist2\(\)](#). Hàm này trả ra tất cả các khoảng cách của các cặp cột của 2 ma trận. Dùng để tính nearest neighbor khi sử dụng sift feature.

Yêu cầu nộp:

- File báo cáo với các mô tả về cách thực hiện các việc được giao và kết quả.
- Mã nguồn.