Principles of Data Science Project 3
Feature Encoding

Hongzhou Liu
517030910214
deanlhz@sjtu.edu.cn

Xuanrui Hong
517030910227
hongxuanrui.1999@sjtu.edu.cn

Qilin Chen
517030910155
1017856853@sjtu.edu.cn

***Abstract*—Hello**
***Index Terms*—something**

I. Introduction

A. SIFT

Scale-invariant feature transform (SIFT) is a machine vision algorithm used to detect and describe local features in an image. It looks for extreme points in the spatial scale and extracts its position, scale, rotation invariant. This algorithm was published by David Lowe in 1999, and was summarized in 2004. [1] The description and detection of local image features can help identify objects. SIFT features are based on some local appearance points of interest on the object and are not related to the size and rotation of the image. The tolerance to light, noise, and slight changes in viewing angle is also quite high. Based on these characteristics, they are highly conspicuous and relatively easy to capture. In a huge feature database, objects are easy to identify and rarely misidentified.

The main steps of the sift algorithm are as follows:

1) Scale-space extrema detection: The images are convolved with Gaussian filters at different scales, and then continuous Gaussian blur is used to blur the image differences to find the key points. The key point is based on the maximum and minimum Gaussian difference (DoG) at different scales. In other words, the D(x,y,) of the DoG image is caused by:

$$D(x, y, \sigma) = L(x, y, k_i\sigma) - L(x, y, k_j\sigma) \tag{1}$$

$L(x, y, k\sigma)$ is the convolution of the original image $I(x, y)$ and Gaussian blur $G(x, y, k\sigma)$ under the condition of the scale $k\sigma$, for example:

$$L(x, y, \sigma) = G(x, y, k\sigma) \times I(x, y) \tag{2}$$

G(x,y,k) is a variable-scale Gaussian function:

$$G(x, y, \sigma) = \frac{1}{2\Pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2} \tag{3}$$

Once the DoG image is obtained, the maximum and minimum values in the DoG image can be found as key points. In order to determine the key points, each pixel in the DoG image will be made with eight pixels around the center of itself, and nine pixels in the same position of the adjacent scale magnification in the same group of DoG images, for a total of 26 points. For comparison, if this pixel is the maximum and minimum of these twenty-six pixels, this pixel is called a key point.

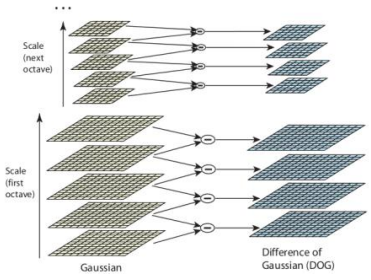


Fig. 1.

2) Keypoint localization: There may be too many key points in different size spaces, and some key points may be relatively difficult to identify or susceptible to noise interference. The next step of the SIFT algorithm will locate each key point by the information of pixels near the key point, the size of the key point, and the main curvature of the key point, thereby eliminating the key points that are located on the side or are susceptible to noise

3) Orientation assignment: After the above steps, feature points that exist at different scales have been found. In order to achieve image rotation invariance, the direction of the feature points needs to be assigned. Use the gradient distribution characteristics of the pixels in the neighborhood of the feature point to determine its direction parameters, and then use the gradient histogram of the image to find the stable direction of the local structure of the key point.

4) Keypoint descriptor: Through the above steps, the location, scale and direction information of SIFT feature points have been found. Next, use a set of vectors to describe the key points, that is, to generate feature point descriptors. There are roughly three steps in generating feature descriptors:

- Correct the main direction of rotation to ensure rotation invariance.
- Generate descriptors and ultimately form a 128-dimensional feature vector
- In the normalization process, the feature vector length is normalized to further remove the influence of lighting.

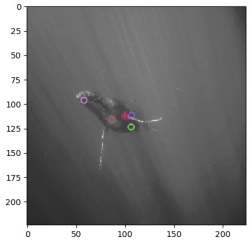
II. Experiments

A. Sift descriptors

We used sift to extract the local descriptors of the images, and then used BOW, VLAD, FV to encode the features, and

(a) original image

(b) processed image



(c) image after sift

Fig. 2. humpback

finally put them into the SVM for classification. As shown in fig. 2, when using sift to extract local descriptors, we encountered some images that could not be extracted. We improved the image contrast to processe these images, and finally got the results.

1) BOW: In this experiment, we used the BOW model to encode the descriptors extracted by sift, and set different clusters k in $[8, 16, 32, 64, 128, 256, 512, 1024]$. Besides, we normalized the feature vectors by z-score. After that, we fed encoded feature vectors into SVM for image classification and chose two different kernels of linear and rbf with the change of C.

Our experiment results are shown in Tab. I. We can find that as k increases, the experimental results get better. We deem the reason is that when k is larger, the bag of words after codebook construction is larger, so the difference among the different types of images after feature encoding is also greater. Besides, we can find that rbf kernel performs better than linear kernel. We think this is because the dimension of the feature vector obtained by the BOW model is k, and rbf kernel is more suitable for classification with fewer feature dimensions than linear kernel. From the results we can see BOW'performance is poor because its highest accuracy is only 0.2488. It is obvious because when mapping, BOW uses the bag of words to quantify the image features for constructing a word frequency histogram. And the word frequency histogram is the encoded feature vector, so much information is lost.

TABLE I
ACCURACY OF STFT FEATURES BASED ON BOW(Z-SCORE) MODEL

Acc. / M / k	SIFT + BOW + SVM			
	C	linear kernel	C	rbf kernel
8	0.0005	0.0699	0.5	0.1361
8	0.001	0.0854	1.0	0.1394
8	0.005	0.1138	5.0	0.1379
8	0.01	0.1213	10	0.1369
16	0.0005	0.0882	0.5	0.1744
16	0.001	0.1134	1.0	0.1786
16	0.005	0.1470	5.0	0.1823
16	0.01	0.1580	10	0.1763
32	0.0005	0.1183	0.5	0.1953
32	0.001	0.1401	1.0	0.2018
32	0.005	0.1767	5.0	0.1963
32	0.01	0.1826	10	0.1885
64	0.0005	0.1495	0.5	0.2151
64	0.001	0.1723	1.0	0.2205
64	0.005	0.2049	5.0	0.2145
64	0.01	0.2103	10	0.2059
128	0.0005	0.1774	0.5	0.2257
128	0.001	0.2037	1.0	0.2342
128	0.005	0.2210	5.0	0.2260
128	0.01	0.2218	10	0.2211
256	0.0005	0.2054	0.5	0.2291
256	0.001	0.2262	1.0	0.2434
256	0.005	0.2321	5.0	0.2351
256	0.01	0.2255	10	0.2317
512	0.0005	0.2293	0.5	0.2242
512	0.001	0.2390	1.0	0.2444
512	0.005	0.2272	5.0	0.2400
512	0.01	0.2133	10	0.2359
1024	0.0005	0.2408	0.5	0.2231
1024	0.001	**0.2440**	1.0	0.2511
1024	0.005	0.2164	5.0	**0.2488**
1024	0.01	0.2036	10	0.2473

2) VLAD: In this experiment, we used the VLAD model to encode the descriptors extracted by sift. The dimension of VLAD feature vector is $k * d$, where k is the cluster number and d is the dimension of each descriptor which equals 128. Because the feature dimension is too high, we use LDA and PCA for feature reduction. Besides, the experiments of VLAD consume time and computing resources, we can't set many different cluster numbers and large cluster numbers. So we only set k in $[4, 8, 16]$ for experiments. After feature encoding and reduction, we fed encoded feature vectors into SVM for image classification and chose two different kernels of linear and rbf with the change of C.

Our experiment results are shown in Tab. II, Tab. III. Generally speaking, it performs better when k is smaller. We deem the reason is that the larger the cluster number, the higher the feature dimension, and the more information is lost after dimensionality reduction. From the results we can find that linear kernel and rbf kernel perform similarly after LDA dimensionality reduction, but linear kernel performance

TABLE II
ACCURACY OF STFT FEATURES BASED ON VLAD MODEL AFTER LDA

Acc. / M / k	SIFT + VLAD + SVM + LDA			
	C	linear kernel	C	rbf kernel
4	0.0005	0.2202	0.5	0.2722
4	0.001	0.2524	1.0	**0.2724**
4	0.005	0.2669	5.0	0.2544
4	0.01	**0.2670**	10	0.2493
8	0.0005	0.2310	0.5	0.2645
8	0.001	0.2506	1.0	0.2649
8	0.005	0.2590	5.0	0.2519
8	0.01	0.2578	10	0.2462
16	0.0005	0.2459	0.5	0.2517
16	0.001	0.2542	1.0	0.2517
16	0.005	0.2557	5.0	0.2451
16	0.01	0.2540	10	0.2399

TABLE III
ACCURACY OF STFT FEATURES BASED ON VLAD MODEL AFTER PCA

Acc. / M / k	SIFT + VLAD + SVM + PCA			
	C	linear kernel	C	rbf kernel
4	0.0005	0.2389	0.5	0.0915
4	0.001	0.2476	1.0	0.1574
4	0.005	0.2513	5.0	**0.1633**
4	0.01	0.2513	10	0.1628
8	0.0005	0.2437	0.5	0.0590
8	0.001	0.2499	1.0	0.0809
8	0.005	0.2491	5.0	0.0886
8	0.01	0.2489	10	0.0888
16	0.0005	0.2644	0.5	0.0509
16	0.001	**0.2660**	1.0	0.0627
16	0.005	0.2646	5.0	0.0683
16	0.01	0.2625	10	0.0683

TABLE IV
ACCURACY OF STFT FEATURES BASED ON FV MODEL AFTER LDA

Acc. / M / k	SIFT + FV + SVM + LDA			
	C	linear kernel	C	rbf kernel
4	0.0005	0.2504	0.5	0.2696
4	0.001	0.2661	1.0	**0.2703**
4	0.005	0.2688	5.0	0.2589
4	0.01	**0.2689**	10	0.2543
8	0.0005	0.2494	0.5	0.2578
8	0.001	0.2594	1.0	0.2561
8	0.005	0.2607	5.0	0.2468
8	0.01	0.2560	10	0.2448
16	0.0005	0.2312	0.5	0.2334
16	0.001	0.2351	1.0	0.2280
16	0.005	0.2336	5.0	0.2198
16	0.01	0.2315	10	0.2190

TABLE V
ACCURACY OF STFT FEATURES BASED ON FV MODEL AFTER PCA

Acc. / M / k	SIFT + FV + SVM + PCA			
	C	linear kernel	C	rbf kernel
4	0.0005	0.2420	0.5	0.0476
4	0.001	0.2489	1.0	0.0829
4	0.005	0.2530	5.0	0.0988
4	0.01	0.2525	10	**0.0988**
8	0.0005	0.2556	0.5	0.0445
8	0.001	0.2606	1.0	0.0447
8	0.005	0.2597	5.0	0.0453
8	0.01	0.2587	10	0.0453
16	0.0005	0.2630	0.5	0.0436
16	0.001	0.2655	1.0	0.0436
16	0.005	0.2655	5.0	0.0436
16	0.01	**0.2658**	10	0.0436

is much better than rbf kernel after PCA dimensionality reduction. We think that the data put into the SVM is linearly separable, and the classification effect of the rbf kernel is greatly affected by the parameters. The parameters we choose are not suitable for data classification after VLAD feature encoding and PCA dimensionality reduction. Overall, VLAD performs better than BOW because it uses the residual of each descriptor with respect to its assigned cluster while BOW only involved simply counting the number of descriptors associated with each cluster in a codebook.

3) Fisher Vector: In this experiment, we used the FV model to encode the descriptors extracted by sift. The dimension of FV feature vector is $2 * k * d$, where k is the cluster number and d is the dimension of each descriptor which equals 128. Because the feature dimension is too high, we use LDA and PCA for feature reduction like the VLAD experiment,. Besides, the experiments of FV also consume time and computing resources, we can't set many different cluster numbers and large cluster numbers. So we only set k in $[4, 8, 16]$ for experiments. After feature encoding and reduction, we fed encoded feature vectors into SVM for image classification and chose two different kernels of linear and rbf with the change of C.

Our experiment results are shown in Tab. IV, V. From the results we can find that FV performs better than BOW and VLAD. This is because Fisher Vector encodes a vector with richer image information which contains 1-order information and 2-order information. Besides in general, as k increases, the experimental results get worse. We deem the reason is that when k is larger, feature vectors resulting from feature encoding contain more redundant information. As for why when using PCA to reduce dimensionality, rbf kernel performs better than linear kernel, we think it is the same as the reason for this phenomenon in VLAD experiment.

III. FUTHER DISCUSSION

1) Impact of scale method on BOW experiment: We know that each dimension of the feature vector obtained by BOW is an integer. To avoid large dimensional differences, we need to standardize the feature vector. In section II-A1, we used Z-Score. In order to get the influence of standardized methods on the experimental results, in this experiment, we used the MaxMin.

Our experiment results are shown in Tab. VI. We can find that Z-Score performs better than MaxMin. We speculate that there may be outliers in the feature vector that affect the experimental results. Therefore Z-Score is more suitable for this project.

2) FV containing only first-order information: We know that FV uses gradient vectors of likelihood functions to encode pictures. In general, it contains both first-order information (expectation) and second-order information (variance). Therefore, in this section, we set the FV to include only first-

TABLE VI
ACCURACY OF STFT FEATURES BASED ON BOW(MAXMIN-SCALE) MODEL

Acc. / M / k	SIFT + BOW + SVM			
	C	linear kernel	C	rbf kernel
8	0.0005	0.0430	0.5	0.0959
8	0.001	0.0430	1.0	0.1080
8	0.005	0.0430	5.0	0.1208
8	0.01	0.0436	10	0.1247
16	0.0005	0.0435	0.5	0.0965
16	0.001	0.0435	1.0	0.1136
16	0.005	0.0435	5.0	0.1428
16	0.01	0.0450	10	0.1556
32	0.0005	0.0456	0.5	0.0874
32	0.001	0.0456	1.0	0.1154
32	0.005	0.0464	5.0	0.1619
32	0.01	0.0557	10	0.1773
64	0.0005	0.0440	0.5	0.0760
64	0.001	0.0440	1.0	0.1063
64	0.005	0.0486	5.0	0.1711
64	0.01	0.0608	10	0.1870
128	0.0005	0.0442	0.5	0.0636
128	0.001	0.0442	1.0	0.0977
128	0.005	0.0556	5.0	0.1708
128	0.01	0.0743	10	0.1904
256	0.0005	0.0433	0.5	0.0587
256	0.001	0.0433	1.0	0.0826
256	0.005	0.0657	5.0	0.1730
256	0.01	0.0975	10	0.2006
512	0.0005	0.0468	0.5	0.0523
512	0.001	0.0468	1.0	0.0756
512	0.005	0.0916	5.0	0.1738
512	0.01	0.1345	10	**0.2071**
1024	0.0005	0.0430	0.5	0.0439
1024	0.001	0.0442	1.0	0.0568
1024	0.005	0.0568	5.0	0.1399
1024	0.01	**0.1418**	10	0.1772

order information, and conduct a comparative experiment with section II-A3.

TABLE VII
ACCURACY OF STFT FEATURES BASED ON FV(1-ORDER) MODEL AFTER LDA

Acc. / M / k	SIFT + FV + SVM + LDA			
	C	linear kernel	C	rbf kernel
4	0.0005	0.2405	0.5	0.2698
4	0.001	0.2626	1.0	**0.2723**
4	0.005	**0.2702**	5.0	0.2576
4	0.01	0.2678	10	0.2494
8	0.0005	0.2446	0.5	0.2694
8	0.001	0.2610	1.0	0.2662
8	0.005	0.2686	5.0	0.2552
8	0.01	0.2653	10	0.2493
16	0.0005	0.2487	0.5	0.2555
16	0.001	0.2561	1.0	0.2538
16	0.005	0.2550	5.0	0.2471
16	0.01	0.2515	10	0.2428

Our experiment results are shown in Tab. VII, VIII. From the results we can find that FV containing only first-order information even performs slightly better than FV containing first- and second-order information. From this we can deduce that sometimes FV can only use first-order information for feature encoding, which also reduces the requirements for computing resources. Besides, we can find that also containing only first-order information, FV performs better than VLAD.

TABLE VIII
ACCURACY OF STFT FEATURES BASED ON FV MODEL AFTER PCA

Acc. / M / k	SIFT + FV + SVM + PCA			
	C	linear kernel	C	rbf kernel
4	0.0005	0.2440	0.5	0.1214
4	0.001	0.2496	1.0	0.1659
4	0.005	0.2507	5.0	0.1733
4	0.01	0.2473	10	**0.1733**
8	0.0005	0.2496	0.5	0.0588
8	0.001	0.2560	1.0	0.1022
8	0.005	0.2568	5.0	0.1154
8	0.01	0.2558	10	0.1154
16	0.0005	0.2593	0.5	0.0445
16	0.001	**0.2615**	1.0	0.0599
16	0.005	0.2613	5.0	0.0702
16	0.01	0.2604	10	0.0702

We think the reason is that GMM clustering can extract feature information better than K-Means clustering.

REFERENCES

[1] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proceedings of the seventh IEEE international conference on computer vision*, vol. 2. Ieee, 1999, pp. 1150–1157.