第四次作业：基于 BOW 和 VLAD 的图像检索实验报告

■ BOW

（1） MATLAB 代码

```
%% BOW
% 清理空间
    clear;
    close all;
    clc;

%% Read xxxx.SIFT
% 读入 SIFT 特征
    num_pic = 1000;
    KM = 1000;
    num_sample = 6000 * 15;

    srcFolderPath = './SIFT';
    allFiles = dir(srcFolderPath);
    imgCount = 0;
    for i = 3 : length(allFiles)
        fileName = allFiles(i).name;
        if length(fileName) > 3 && strcmp(fileName(end-9 : end-6), '.jpg') == 1
            imgCount = imgCount + 1;
            imgPath = [srcFolderPath, '/', fileName(1:end-6)];
            %fprintf('File %d: %s\n', imgCount, imgPath);
            sift = readsift(imgPath);
            SiftFeat{imgCount} = sift;
        end
    end

%% SIFT_ALL
    sift_all = [];

    for i=1:num_pic
        sift_all = [sift_all;SiftFeat{i}];
    end

% load 'sift_all'

% 抽取部分 SIFT 特征
    [sm,sn] = size(sift_all);
    ind_chose = randperm(sm,num_sample);
```

```
    sift_chose = sift_all(ind_chose,:);

% KMEANS 获得码本
    num_codebook = KM;
    [Idx_all,C] = kmeans(sift_chose,num_codebook,'MaxIter',100,'display','iter');

    clear sift_all;clear SiftFeat;

%% BOW
% SIFT 特征读入
    srcFolderPath = './SIFT';
    allFiles = dir(srcFolderPath);
    imgCount = 0;
    for i = 3 : length(allFiles) %从 3 开始
        fileName = allFiles(i).name;
        if length(fileName) > 3 && strcmp(fileName(end-9 : end-6), '.jpg') == 1 % find JPG image
file
            imgCount = imgCount + 1;
            imgPath = [srcFolderPath, '/', fileName(1:end-6)];
            %fprintf('File %d: %s\n', imgCount, imgPath);
            sift = readsift(imgPath); % read sift--> 读取存储的 sift 算子
            [sm,sn]=size(sift);
            SiftFeat{imgCount} = sift(randperm(sm, floor(sm)),:);
            %传入归一化后的 sift,  而且已经被转制并随机抽取部分行
        end
    end

% 计算到中心聚类点的距离，根据距离确定直方图
    for i = 1:num_pic
        % disp(i)
        similarDistances = pdist2(SiftFeat{i},C);
        [minElements,idx] = min(similarDistances,[],2);
        bins = 0.5:1:KM+0.5;
        hist = histogram(idx,bins);
        Features = hist.Values;
        % L1 归一化
        Features = Features./sum(Features);
        %       % L2 归一化
        %       Features = Features./sqrt(sum(Features.^2));
        Final_Hist(i,:) = Features';
    end

% 计算各图片与其它图片间的距离
    result_Ech = zeros(num_pic,num_pic);
```

```
    for i=1:num_pic
        for j=1:num_pic
            result_Ech(i,j) = sum((Final_Hist(i,:)-Final_Hist(j,:)).^2);
        end
    end

% 提取距离最小的前四幅图的 id
    temp_result = sort(result_Ech,2);
    result = zeros(num_pic,4);
    for i=1:num_pic
        for j=1:4
            temp_find = find(result_Ech(i,:)==temp_result(i,j));
            result(i,j) = temp_find(1);
        end
    end

% 计算相关程度
    for i=1:(num_pic/4)
        class{i} = [(i-1)*4+1:(i-1)*4+4];
    end

    for i=1:num_pic
        num_result = 1;
        for j=2:4
            temp = find(class{ceil(i/4)}==result(i,j));
            if ~isempty(temp)
                num_result = num_result+1;
            end
            result_one(i) = num_result;
        end
    end

% 输出检索准确率
    result_average = mean(result_one)./4;
    BOW1000L1 = result_average;
    str_name = ['BOW',num2str(KM),'L1'];
    save(str_name,str_name);
```

(2) 实验结果分析

为便于计算，在 BOW 实验中，码本数目分别取 1000,2000,3000,5000,6000,随机抽取 SIFT 样本数目为 90000,并考虑使用 L1、L2 范数所带来的结果如图 1 所示。从图 1 可以看到，在 SIFT 特征采样数量相同的情况下，检索准确率随着码本数量增大而减小，

且 L2 范数的检索效果优于 L1 范数的检索效果。

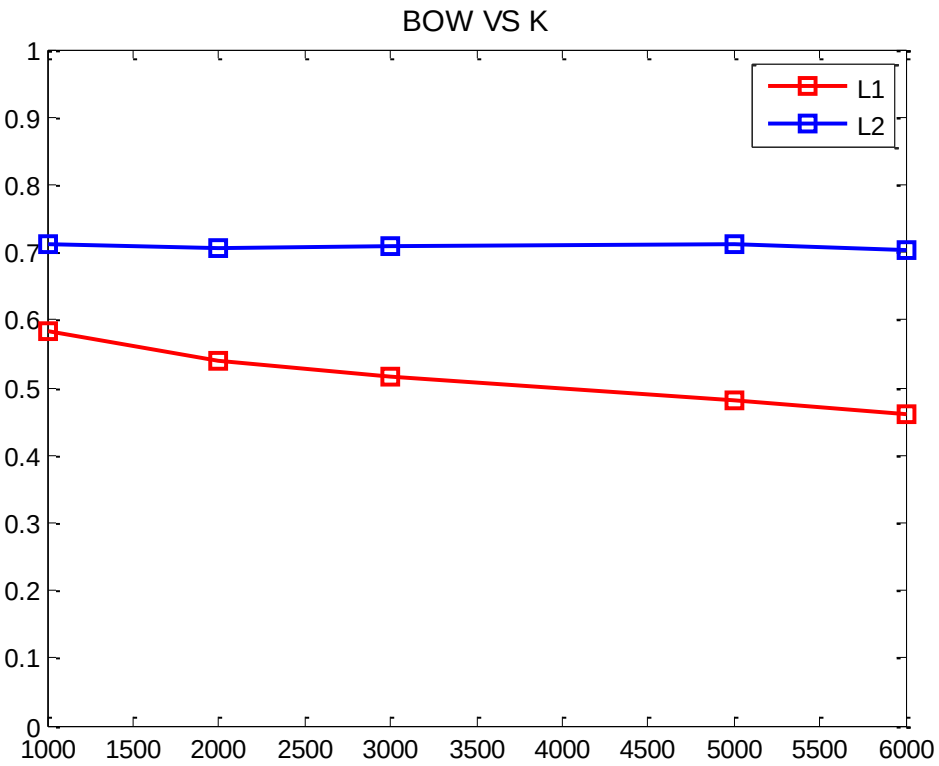

图 1　BOW 检索率随码本数量及范数的变化图

■ VLAD

(1) MATLAB 代码

```
%% VLAD
% 清理空间
    clear;
    close all;
    clc;

%% SIFT
% 读取
    srcFolderPath = './SIFT';
    allFiles = dir(srcFolderPath);
    imgCount = 0;
    for i = 3 : length(allFiles) %从 3 开始
        fileName = allFiles(i).name;
        if length(fileName) > 3 && strcmp(fileName(end-9 : end-6), '.jpg') ==
 1 % find JPG image file
            imgCount = imgCount + 1;
            imgPath = [srcFolderPath, '/', fileName(1:end-6)];
```

```
            fprintf('File %d: %s\n', imgCount, imgPath);
            sift = readsift(imgPath); % read sift--> 读取存储的 sift 算子
            [sm,sn]=size(sift);
            SiftFeat{imgCount} = sift(randperm(sm, floor(sm*1)),:);
           %传入归一化后的 sift,  而且已经被转制并随机抽取 300 行
        end
    end

%% KMEANS
% 提取码本
    sift_all = [];
    num_pic=1000;
    KM=8;

    for i=1:num_pic
        sift_all = [sift_all;SiftFeat{i}];
    end
    [Idx_all,C] = kmeans(sift_all,KM,'MaxIter',100,'display','iter');

    size_sift=zeros(1,num_pic);
    for i=1:num_pic
         m = size(SiftFeat{i});
         size_sift(i) = m(1);
    end
    sum_size = 1;
    for i=1:num_pic
        temp_Idx = Idx_all(sum_size:sum_size+size_sift(i)-1);
        sum_size = sum_size+size_sift(i);
        Idx{i} = temp_Idx;
    end

%% 计算每幅图和聚点的累计残差
    for i=1:num_pic
        temp_v = zeros(KM,128);
        temp_sift = SiftFeat{i};
        for num_1=1:size_sift(i)
            temp_v(Idx{i}(num_1),:)=temp_v(Idx{i}(num_1),:)
                                +temp_sift(num_1,:)-C(Idx{i}(num_1),:);
        end
        v_chuan = [];
        for num_2=1:KM
            v_chuan = [v_chuan,temp_v(num_2,:)];
        end
        v_chuan = bsxfun(@times, v_chuan, 1./sqrt(sum(v_chuan.^2,2)));
```

```
        v{i}= v_chuan;
    end
%% 计算每幅图累计残差的 L2 距离
    result_Ech = zeros(num_pic,num_pic);
    for i=1:num_pic
        for j=1:num_pic
            result_Ech(i,j) = sum((v{i}-v{j}).^2);
        end
    end

%% 提取距离最小的前四幅图
    temp_result = sort(result_Ech,2);
    result = zeros(num_pic,4);
    for i=1:num_pic
        for j=1:4
            temp_find = find(result_Ech(i,:)==temp_result(i,j));
            result(i,j) = temp_find(1);
        end
    end

%% 计算前四幅图相关程度
    for i=1:(num_pic/4)
        class{i} = [(i-1)*4+1:(i-1)*4+4];
    end

    for i=1:num_pic
        num_result = 1;
        for j=2:4
            temp = find(class{ceil(i/4)}==result(i,j));
            if ~isempty(temp)
                num_result = num_result+1;
            end
            result_one(i) = num_result;
        end
    end

%% 输出结果
result_average = mean(result_one)./4
```

(2) 实验结果及分析

在 VLAD 实验中，码本数目分别取 8,16,32,64，使用全部的 SIFT 特征,并考虑使用 L2 范数归一化，结果如图 2 所示。从图 2 可以看到，在 SIFT 特征采样数量相同的情况下，检索准确率随着码本数量增大而增大。

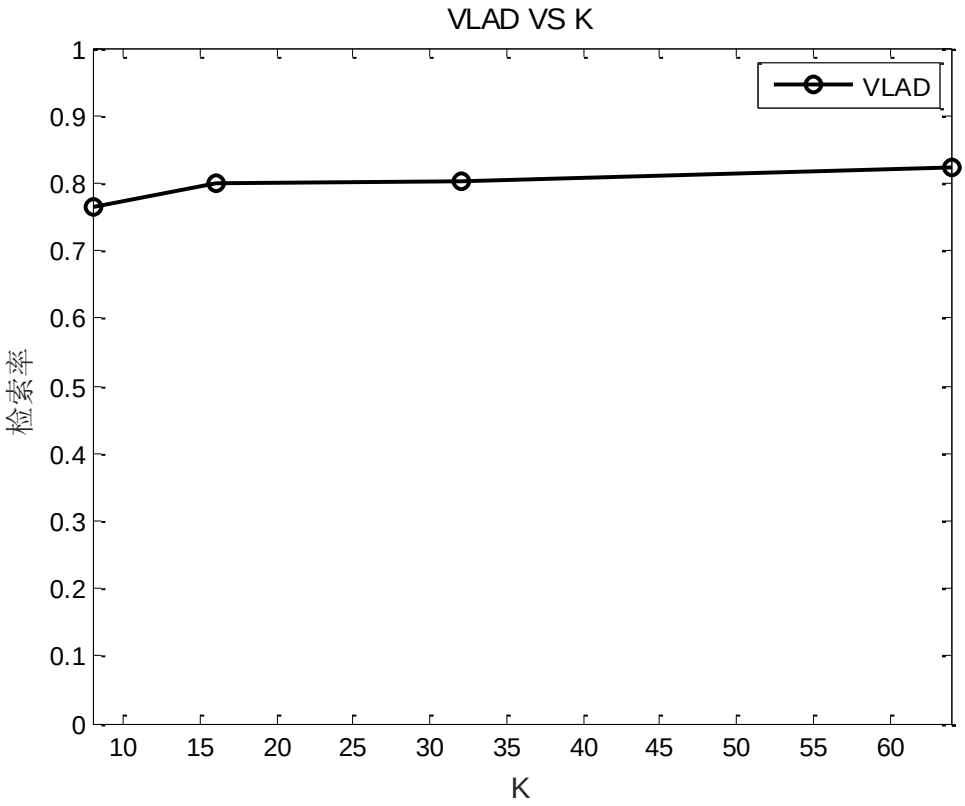

图 2　VLAD 检索率随码本数量及范数的变化图