高级计算机视觉实验报告

ZYZ

2019 年 5 月 16 日

1 Project1 - Image Filtering and Hybrid Images

1.1 算法介绍

1.1.1 Image Filtering

1. 根据 filter 的大小，对原图进行 pad ，保证卷积后的输出大小和原图一致；
2. 通过高度、宽度遍历pad后的图片，依次取出和 filter 大小一致的矩阵，对应元素相乘再相加，得到卷积后的值。

1.1.2 Hybrid Images

1. 依次对图1的行列进行高斯滤波，得到低频图像；
2. 图2减去图2的低频图像得到图2的高频图像；
3. 低频与高频相加即为混合图像。

1.2 核心代码

1.2.1 my_imfilter.m

```
function output = my_imfilter(image, filter)
[height, width, channel] = size(image);
[row, column] = size(filter);
image_pad = padarray(image, [(row-1)/2, (column-1)/2], 0, 'both');
output = zeros(height, width, channel);
```

```
for i = 1:(height)
    for j = 1:(width)
        for k = 1:3
            temp = image_pad(i:i+row-1, j:j+column-1, k);
            conv = temp.*filter;
            output(i, j, k) = sum(conv(:));
        end
    end
end
```

1.2.2 proj1.m

```
low_frequencies = my_imfilter(image1, filter);
low_frequencies = my_imfilter(low_frequencies, filter');
high_frequencies = image2 - my_imfilter(my_imfilter(image2,
    filter), filter');
hybrid_image = low_frequencies + high_frequencies;
```

1.3 实验结果

1.3.1 Image Filtering

image 为 marilyn.bmp

图 1: identity image

图 2: blur image

图 3: large blur image

图 4: sobel image

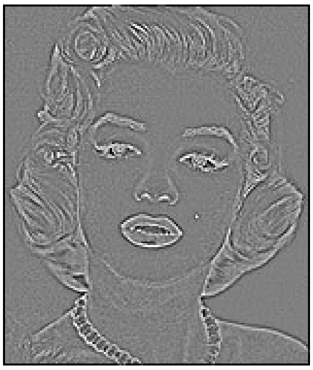

图 5: laplacian image

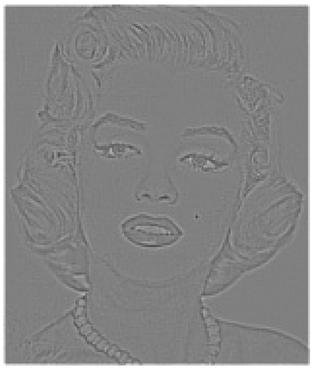

图 6: high pass image

1.3.2 Hybrid Images

image1 为 motorcycle.bmp，image2 为 bicycle.bmp

图 7: 低频图像

图 8: 高频图像

图 9: 混合图像

图 10: 不同尺度的混合图像

2 Project2 - Local Feature Matching

2.1 算法介绍

Interest point detection

建立 Harris 角点检测算法，在图像上发现角点特征，其具有旋转不变性。根据泰勒级数一阶展开近似计算，得到 Harris 矩阵公式。由矩阵的行列式和迹得到特征值的相关式子，再计算角点响应值，大于阈值的认为是特征点。

Local feature description

对 filter 求导，再分别在 x、y 方向对图片做卷积运算，基于图像局部的梯度方向，分配给每个关键点位置一个或多个方向。所有后面的对图像数据的操作都相对于关键点的方向、尺度和位置进行变换，从而提供对于这些变换的不变性。在每个关键点周围的邻域内，在选定的尺度上测量图像局部的梯度。

Feature matching

将关键点附近的领域划分为4×4个子区域，每个子区域有8个方向。然后将领域内的采样点分配到对应的子区域内，将子区域内的梯度值分配到8个方向上，计算其权值。遍历所有采样点，得到4*4*8维的一个特征算子。进行归一化处理使其对亮度变化不敏感。

2.2 核心代码

2.2.1 get_interesting_points.m

```
filter_gauss = fspecial('Gaussian', [5 7], 3);
gauss2 = fspecial('Gaussian', [30 30], 1);
image_new = imfilter(image, filter_gauss);

[der_x,der_y] = imgradientxy(image_new,'sobel');
der_xx = imfilter(der_x.*der_x, gauss2,'same');
der_yy = imfilter(der_y.*der_y, gauss2,'same');
der_xy = imfilter(der_x.*der_y, gauss2,'same');

alpha=0.06;
r = der_xx.*der_yy - der_xy.*der_xy -
    alpha.*(der_xx+der_yy).*(der_xx+der_yy);

threshold = 0.001;
[y,x] = find(r>threshold);
confidence = r(r>0);
```

2.2.2 get_features.m

```
features = zeros(length(x), 128);
gauss_filter = fspecial('Gaussian', [5 7], 1);
[dx_filter,dy_filter] = imgradientxy(gauss_filter);
image_dx = imfilter(image, dx_filter);
image_dy = imfilter(image, dy_filter);

for i = 1:length(x)
    windowx = (x(i)-7):(x(i)+8);
    windowy = (y(i)-7):(y(i)+8);
    gradient_x = image_dx(windowy,windowx);
    gradient_y = image_dy(windowy,windowx);

    theta = atan2(gradient_y,gradient_x);
    theta = radtodeg(theta);
    theta = floor(theta/45);
    theta = theta+4;

    for x1 = 1:4
        for y1 = 1:4
            tangents = theta(x1*4-3:x1*4, y1*4-3:y1*4);
            for bins = 1:8
                feat = eq(tangents,bins);
                features(i,(x1*8 + y1*32) + bins) = sum(sum(feat));
            end
        end
    end
end
features = normc(features);
```

2.2.3 match_features.m

```
matches = [];
dist = pdist2(features1, features2, 'euclidean');
```

```
[sort_dist, index] = sort(dist, 2);
nn_measure = (sort_dist(:,1)./sort_dist(:,2));

threshold = 0.90;
val_threshold(:) = nn_measure(nn_measure < threshold);

inverse_matrix = ones(size(val_threshold,1))*1;
confidences = inverse_matrix./val_threshold;
indices_feature(:) = nn_measure<threshold;

matches(:,1) = find(indices_feature);
matches(:,2) = index(indices_feature, 1);

[confidences, ind] = sort(confidences, 'descend');
matches = matches(ind,:);
```

2.3 实验结果

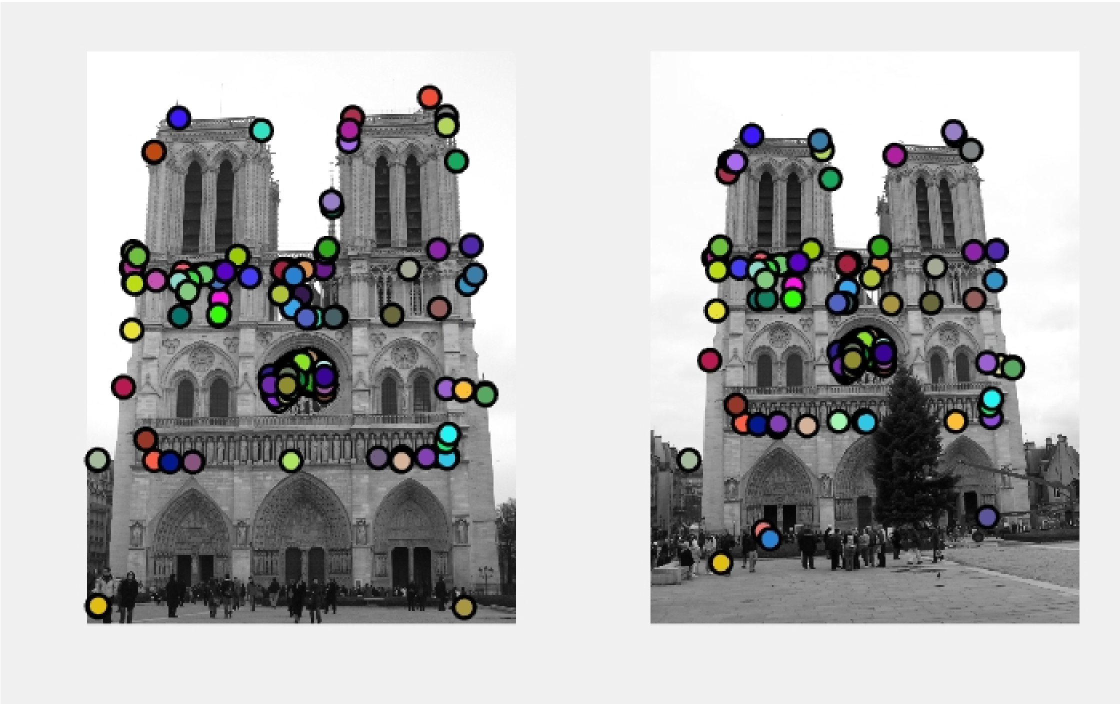

图 11: vis

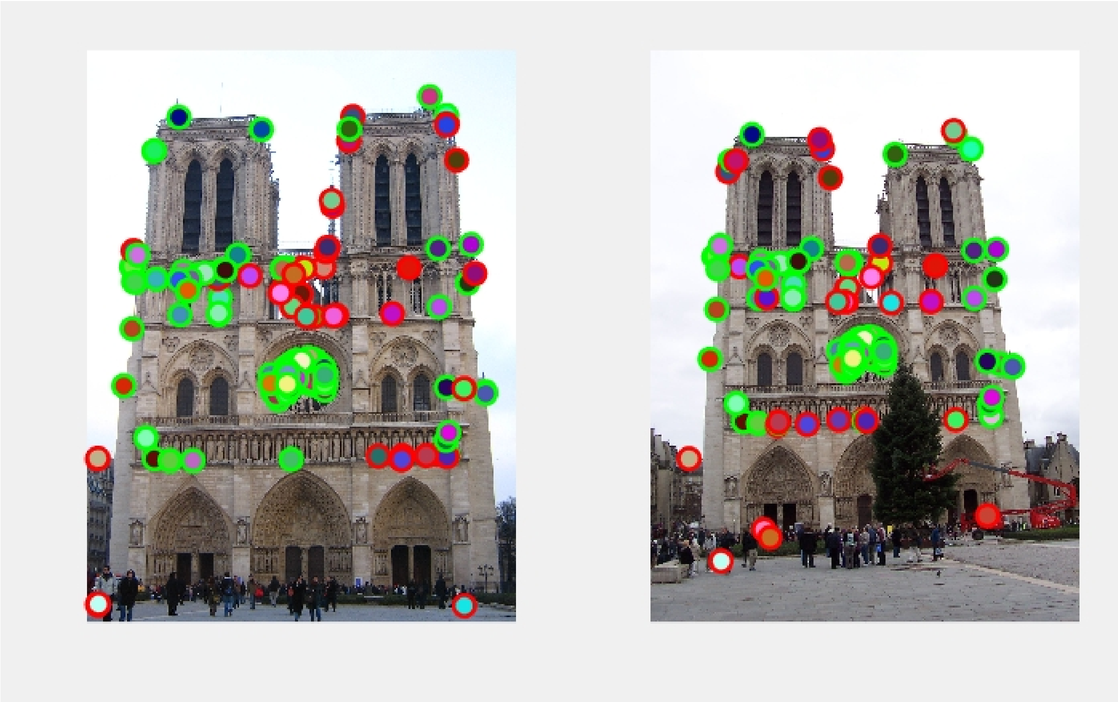

图 12: eval

245 total good matches, 64 total bad matches

3 Project4 - Face detection with a sliding window

3.1 算法介绍

1. 加载训练集的正样本和随机生成的不包含人脸的负样本，并通过调用 vl_hog() 将所有样本转换为 HoG 特征；
2. 在得到的特征上调用 vl_svmtrain() 进行训练，得到一个线性分类器；
3. 训练集上的 training error 应该很小，查看该数据可以辅助检测分类器的正确性；
4. 在测试集上检验分类器的学习情况，从不同尺度上运行分类器，再调用 non_max_supr_bbox() 做非极大值抑制去重。

3.2 核心代码

3.2.1 get_positive_features.m

```
image_files = dir( fullfile( train_path_pos, '*.jpg') ); %Caltech
    Faces stored as .jpg
num_images = length(image_files);

for i = 1:num_images
    consider_image =
        single(imread(fullfile(train_path_pos,image_files(i).name)));
    features_pos(i,:) =
        reshape(vl_hog(consider_image,(feature_params.template_size
        /
        feature_params.hog_cell_size)),1,((feature_params.template_size
        / feature_params.hog_cell_size)^2 * 31));
end
```

3.2.2 get_random_negative_features.m

```
image_files = dir( fullfile( non_face_scn_path, '*.jpg' ));
num_images = length(image_files);
number_samples_image = ceil(num_samples/num_images);

k = 1;
for i = 1:num_images
    consider_image =
        single(rgb2gray(imread(fullfile(non_face_scn_path,image_files(i).name))));
    [y,x] = size(consider_image);
    some_value = 36;
    for j = 1:number_samples_image

        image_cropped =
            consider_image(randi(y-some_value+1)+(0:some_value-1),randi(x-some_value+1)+(0:s
        features_neg(k,:) =
```

```
            reshape(vl_hog(image_cropped,(feature_params.template_size
            /
            feature_params.hog_cell_size)),1,((feature_params.template_size
            / feature_params.hog_cell_size)^2 * 31));

        k = k+1;
    end
end
```

3.2.3 classifier training

```
lambda = 0.0001;
[pos_height,pos_width] = size(features_pos);
[neg_height,neg_width] = size(features_neg);

features_all = [features_pos;features_neg];
feature_labels_pos = ones(pos_height,1);
feature_labels_neg = ones(neg_height,1).*-1;
labels_all = [feature_labels_pos;feature_labels_neg];

[w,b] = vl_svmtrain(features_all',labels_all,lambda);
```

3.2.4 run_detector.m

```
for i = 1:length(test_scenes)

    fprintf('Detecting faces in %s\n', test_scenes(i).name)
    img = imread( fullfile( test_scn_path, test_scenes(i).name ));
    img = single(img)/255;
    if(size(img,3) > 1)
        img = rgb2gray(img);
    end

    cur_bboxes = zeros(0, 4);
```

```
cur_confidences = zeros(0,1);
cur_image_ids = cell(0,1);
value = feature_params.template_size /
    feature_params.hog_cell_size;

for scale = diff_scales
    image_scaled = imresize(img,scale);
    features_test =
        vl_hog(image_scaled,(feature_params.template_size /
        feature_params.hog_cell_size));
    [height,width,depth] = size(features_test);
    for ht = 1:height - value
        for wd = 1:width - value
            image_hog
                =features_test(ht:ht+value-1,wd:wd+value-1,:);
            image_hog_reshape =
                reshape(image_hog,(feature_params.template_size /
                feature_params.hog_cell_size)^2 * 31,1);

            confidence = transpose(w)*image_hog_reshape +b;
            if(confidence > 0.80)
                x1 =
                    (wd-1)*feature_params.hog_cell_size*(1.0/scale)+1;
                y1 =
                    (ht-1)*feature_params.hog_cell_size*(1.0/scale)+1;
                x2 = x1+feature_params.template_size*(1.0 /
                    scale) - 1;
                y2 = y1+feature_params.template_size*(1.0 /
                    scale) - 1;
                bbox=[x1, y1, x2, y2];
                cur_bboxes = [cur_bboxes;bbox];
                cur_confidences = [cur_confidences; confidence];
                cur_image_ids = [cur_image_ids;
                    test_scenes(i).name];

            end
        end
    end
```

```
    end

    [is_maximum] = non_max_supr_bbox(cur_bboxes, cur_confidences,
        size(img));

    cur_confidences = cur_confidences(is_maximum,:);
    cur_bboxes      = cur_bboxes(    is_maximum,:);
    cur_image_ids   = cur_image_ids( is_maximum,:);

    bboxes      = [bboxes;      cur_bboxes];
    confidences = [confidences; cur_confidences];
    image_ids   = [image_ids;   cur_image_ids];
end
```

3.3 实验结果

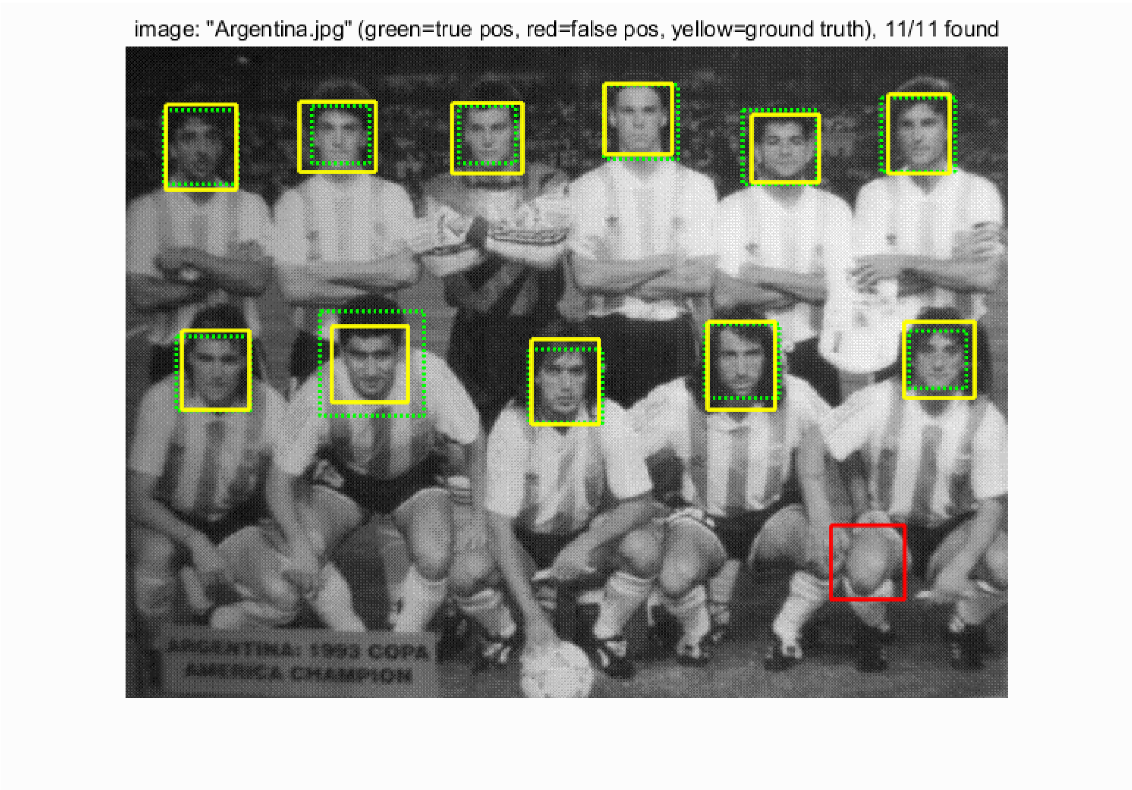

图 13: Argentina

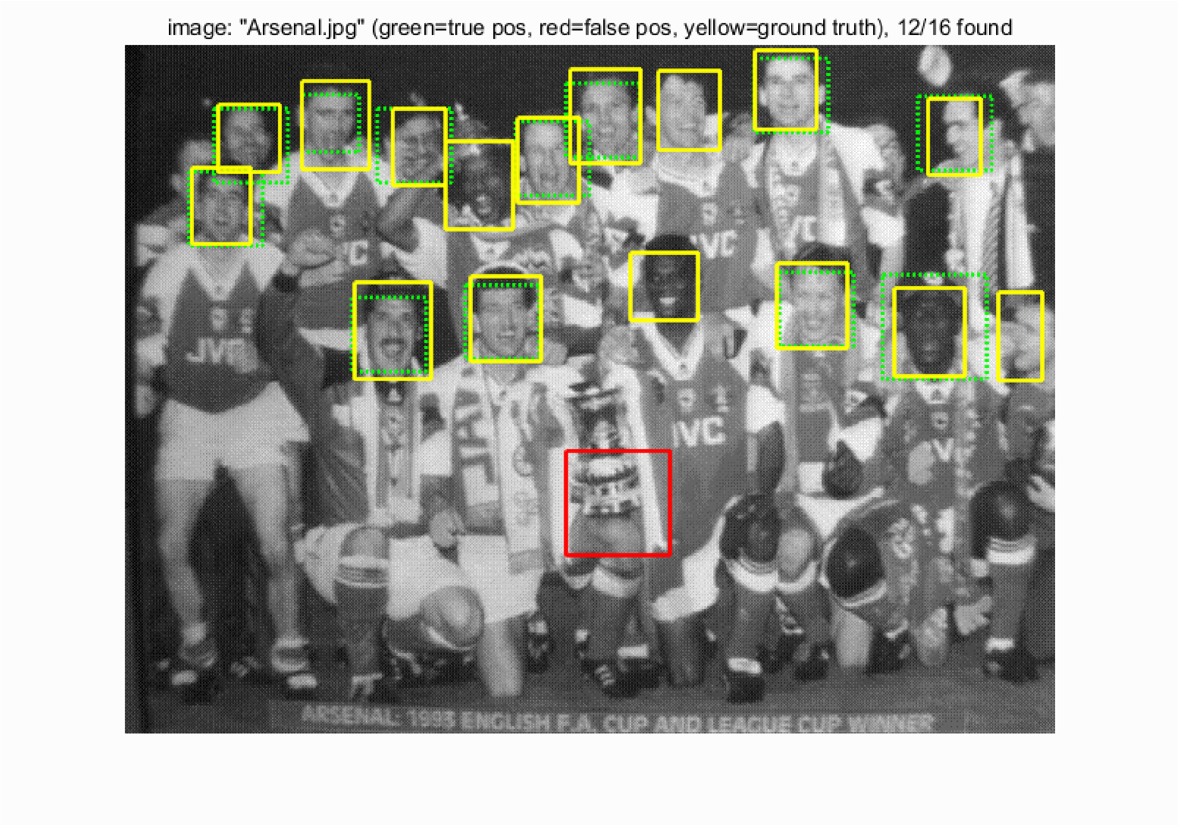
image: "Arsenal.jpg" (green=true pos, red=false pos, yellow=ground truth), 12/16 found
ARSENAL: 1993 ENGLISH F.A. CUP AND LEAGUE CUP WINNER

图 14: Arsenal