心率估计实验报告

热伊莱·图尔贡 2018K8009929030

实验题目：心率估计

- 依据课堂介绍实现利用人脸视频进行心率估计的算法，鼓励自己设计模块

算法各步骤的设计动机

思路：

1. 读取视频
2. 感兴趣区域抠取
3. 原始颜色变化提取
4. 生理信号表示 --颜色空间投影
5. 生理指标提取

- 输入人脸视频，获取感兴趣区域 使用提供材料中的人脸的landmark数据，裁剪出感兴趣的部分，本次实验中采取了整个脸颊部

```
% 输入人脸视频，获取感兴趣区域（脸颊）
function frame = Crop_frame(frame_raw, k, ROI_x, ROI_y)
    % frame_raw 输入视频帧
    % ROI_x 感兴趣区域的 landmark 的横坐标
    % ROI_y 感兴趣区域的 landmark 的纵坐标
    % frame 裁剪出的部分
    global lmk;

    frame_raw_R = frame_raw(:, :, 1);
```

```
    frame_raw_G = frame_raw(:, :, 2);
    frame_raw_B = frame_raw(:, :, 3);

    intrest_region_x = lmk(k, ROI_x);
    intrest_region_y = lmk(k, ROI_y);

    bw = roipoly(frame_raw, intrest_region_x, intrest_region_y);

    frame_R = frame_raw_R .* double(bw);
    frame_G = frame_raw_G .* double(bw);
    frame_B = frame_raw_B .* double(bw);

    frame(:, :, 1) = frame_R;
    frame(:, :, 2) = frame_G;
    frame(:, :, 3) = frame_B;

end
```

- 从感兴趣区域中提取出 RGB 3 通道的颜色变化

```
%从感兴趣区域中提取出 RGB 3 通道的颜色变化
%对整个感兴趣区域的颜色值求平均，以达到降噪的目的
function RGB_Record = Get_RGB(frame)
    % frame 经过裁剪后的视频帧
    % RGB_Record 求解得到的 RGB 颜色值
    [Row, Col, Dep] = size(frame);
    pixel_count = 0;

    R_value = 0.0;
    G_value = 0.0;
    B_value = 0.0;

    for i = 1:Row

        for j = 1:Col

            if frame(i, j, 1) == 0 && frame(i, j, 2) == 0 && frame(i, j, 3) == 0
                continue;
            else
                pixel_count = pixel_count + 1;
                R_value = R_value + double(frame(i, j, 1));
                G_value = R_value + double(frame(i, j, 2));
                B_value = R_value + double(frame(i, j, 3));

            end

        end

    end

    Avg_R_value = R_value / pixel_count;
    Avg_G_value = G_value / pixel_count;
    Avg_B_value = B_value / pixel_count;
```

```
    RGB_Record = [Avg_R_value Avg_G_value Avg_B_value];
end
```

- 通过 RGB 的颜色变化，获得生理信号的表示 从颜色信号获得生理信号选择颜色空间投影法

```
% 通过 RGB 的颜色变化，获得生理信号的表示
% 颜色空间投影
function P = RGB2P(R_value, G_value, B_value, Method, Window)
    global numFrames;
    % R_value R 通道颜色变化
    % G_value G 通道颜色变化
    % B_value B 通道颜色变化
    % Method 数据平滑滤波的方法，本次实验一律采用 Gaussian 滤波
    % Window 平滑滤波的窗口大小
    % P 生理信号
    Smooth_R_value = smoothdata(R_value, Method, Window);
    Smooth_G_value = smoothdata(G_value, Method, Window);
    Smooth_B_value = smoothdata(B_value, Method, Window);

    Avg_Smooth_R_value = sum(Smooth_R_value) / numFrames;
    Avg_Smooth_G_value = sum(Smooth_G_value) / numFrames;
    Avg_Smooth_B_value = sum(Smooth_B_value) / numFrames;

    Normal_R_value = Smooth_R_value ./ Avg_Smooth_R_value;
    Normal_G_value = Smooth_G_value ./ Avg_Smooth_G_value;
    Normal_B_value = Smooth_B_value ./ Avg_Smooth_B_value;

    Avg_Normal_R_value = sum(Normal_R_value) / numFrames;
    Avg_Normal_G_value = sum(Normal_G_value) / numFrames;
    Avg_Normal_B_value = sum(Normal_B_value) / numFrames;

    S1 = 3 .* Normal_R_value - 2 .* Avg_Normal_G_value;
    S2 = 1.5 .* Normal_R_value + Avg_Normal_G_value - 1.5 .* Normal_B_value;

    Var_S1 = var(S1, 1);
    Var_S2 = var(S2, 1);
    alp = Var_S1 / Var_S2;

    P = S1 - alp * S2;
end
```

- 通过生理信号，求解心率 使用时域空间的分析法

```
% 通过生理信号，求解心率
% 时域空间的分析法
function [Golden_HR, HR] = Check(Path, Window)

    global ROI_forehead_x ROI_forehead_y;
    global ROI_left_cheek_x ROI_left_cheek_y;

    ROI_forehead_x = [28 22 21 20 24 23];
```

```
    ROI_forehead_y = ROI_forehead_x + 68;

    ROI_left_cheek_x = [2 3 4 5 32 41];
    ROI_left_cheek_y = ROI_left_cheek_x + 68;

    ROI_right_cheek_x = [13 14 15 16 17 36];
    ROI_right_cheek_y = ROI_right_cheek_x + 68;

    ROI_full_cheek_x = [2 3 4 5 32 31 36 13 14 15 16 47 29 41];
    ROI_full_cheek_y = ROI_full_cheek_x + 68;

    golden_hr_path = strcat(Path, '/HR_gt.txt');
    Golden_HR = textread(golden_hr_path);

    fps_path = strcat(Path, '/fps.txt');
    FPS = textread(fps_path);

    global lmk
    lmk_path = strcat(Path, '/lmk.csv');
    lmk = csvread(lmk_path, 1, 5);

    % 输入视频位置
    video_path = strcat(Path, '/video.avi');
    obj = VideoReader(video_path);

    % 帧的总数
    global numFrames;
    numFrames = obj.NumberOfFrames;

    % 读取每个帧
    for k = 1:numFrames
        % fprintf("Processing The %dth Frame\n", k);

        %读取第k帧
        frame_raw = double(read(obj, k));
        frame_full_cheek = Crop_frame(frame_raw, k, ROI_full_cheek_x,
ROI_full_cheek_y);
        RGB_Record_full_cheek = Get_RGB(frame_full_cheek);
        R_value(k) = RGB_Record_full_cheek(1);
        G_value(k) = RGB_Record_full_cheek(2);
        B_value(k) = RGB_Record_full_cheek(3);

    end

    P = RGB2P(R_value, G_value, B_value, 'gaussian', Window);
    % 以极大值点为一个周期的波峰，求解心率
    [Pks, Locs] = findpeaks(P);
    numLocalMax = size(Locs);

    for i = 1:numLocalMax(2) - 1
        Distance(i) = Locs(i + 1) - Locs(i);
    end

    Avg_Distance = sum(Distance) / (numLocalMax(2) - 1);
    HR = uint8(FPS / Avg_Distance * 60);

end
```

- 生理指标提取

```
num = 8;
% 人为选取的窗口
Windows = [50, 50, 80, 70, 15, 30, 30, 15];
% 视频帧率
% Windows = [61, 61, 61, 61, 25, 30, 30, 25];

for i = 1:num
    fprintf("Processing The %dth Video\n", i);
    Path = strcat('../videos/video', num2str(i));
    [Golden_HR, HR] = Check(Path, Windows(i));
    Golden_Trace(i) = Golden_HR;
    Test_HR(i) = HR;
end

fprintf("\n=======================Results======================\n");

for i = 1:num
    fprintf("Video %d:\n", i);
    fprintf("Golden Heart Rate: %d\n", Golden_Trace(i));
    fprintf("Test Heart Rate: %d\n", Test_HR(i));
    fprintf("\n");
end
```

实验结果分析（包括可视化）

- 人为选取平滑窗口大小以逼近真实心率时的测试结果：

视频	Ground-Truth HR	Test HR
Video1	74	70
Video2	64	70
Video3	52	57
Video4	64	66
Video5	87	90
Video6	73	71
Video7	63	61
Video8	92	90

```
Processing The 1th Video
Processing The 2th Video
Processing The 3th Video
Processing The 4th Video
Processing The 5th Video
Processing The 6th Video
Processing The 7th Video
Processing The 8th Video

=======================Results======================
Video 1:
Golden Heart Rate: 74
Test Heart Rate: 70

Video 2:
Golden Heart Rate: 64
Test Heart Rate: 70

Video 3:
Golden Heart Rate: 52
Test Heart Rate: 57

Video 4:
Golden Heart Rate: 64
Test Heart Rate: 66

Video 5:
Golden Heart Rate: 87
Test Heart Rate: 90

Video 6:
Golden Heart Rate: 73
Test Heart Rate: 71

Video 7:
Golden Heart Rate: 63
Test Heart Rate: 61

Video 8:
Golden Heart Rate: 92
Test Heart Rate: 90
```

- 使用视频帧率替代人为选取的平滑窗口的大小时的测试结果：

视频	Ground-Truth HR	Test HR
Video1	74	74
Video2	64	77
Video3	52	77
Video4	64	87
Video5	87	62
Video6	73	71
Video7	63	61

视频	Ground-Truth HR	Test HR
Video8	92	41

```
Processing The 1th Video
Processing The 2th Video
Processing The 3th Video
Processing The 4th Video
Processing The 5th Video
Processing The 6th Video
Processing The 7th Video
Processing The 8th Video

=======================Results==========
Video 1:
Golden Heart Rate: 74
Test Heart Rate: 74

Video 2:
Golden Heart Rate: 64
Test Heart Rate: 77

Video 3:
Golden Heart Rate: 52
Test Heart Rate: 77

Video 4:
Golden Heart Rate: 64
Test Heart Rate: 87

Video 5:
Golden Heart Rate: 87
Test Heart Rate: 62

Video 6:
Golden Heart Rate: 73
Test Heart Rate: 71

Video 7:
Golden Heart Rate: 63
Test Heart Rate: 61

Video 8:
Golden Heart Rate: 92
Test Heart Rate: 41
```

观察实验测试结果，人为选取平滑窗口大小以逼近真实心率时的测试得到的心率与Ground-Truth HR的误差都限制在10以内，平均误差3.25，取得了较好的实验效果。 使用视频帧率替代人为选取的平滑窗口的大小时的测试结果比人为选取大。

算法优缺点、适用范围分析

- 感兴趣区域抠取： 选择了脸颊而非额头，导致测试结果比起额头误差大一些。 使用人脸的landmark时，额头部分只有眉毛作为下边缘，而没有上边缘； 有的视频，额头部分有头发遮盖。 因此综合考虑后，选取了脸颊部分进行计算

- 对采样的得到的数据进行平滑操作： 在movmean，movmedium，gaussian，sgolay 4种数据平滑方法中经过测试对比选择采用gaussian。

- 从感兴趣区域提取出原始颜色变化这一阶段（Get_RGB）产生误差： 由于对于感兴趣区域的提取不够精准，胡子、眉毛等部位可能会造成影响。

- 使用时域空间的分析法从生理信号获得心率： 可以获得平均心率，也可以估算瞬时心率。但是少了消除误差的处理。 可以采用逐差法减少误差来优化代码，本次实验没有进行优化处理（使用了直接求平均）