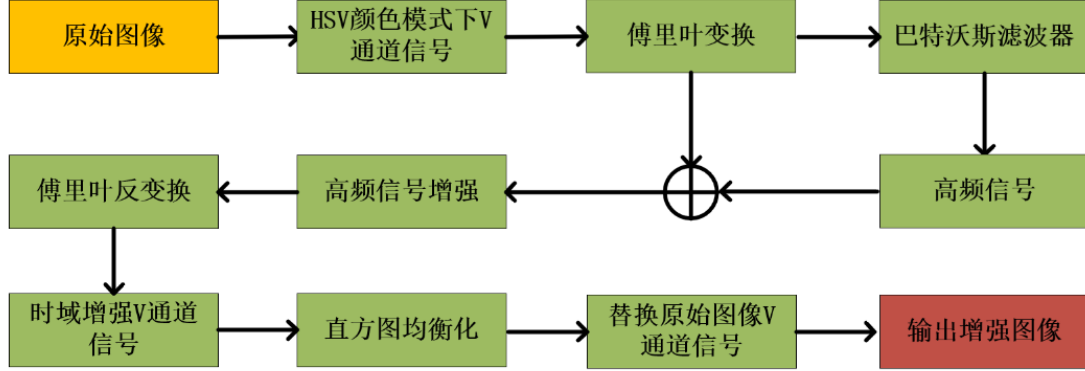


一、整体设计思路

对于要求中的眼科图像,需要通过算法使得眼球(包括眼黑和眼白)纹理信息更加清晰,同时避免失真和过曝问题。由于眼球的纹理细节为图像高频信号,为了增强图像,容易联想到抑制低频噪声,增强高频信号,并通过增强图像亮度和对比度等方式实现要求。本实验整体设计思路下图所示。



首先根据原始图像,将其转换为 HSV 颜色模式下的信号,然后将 V(Value,亮度)通道的信号提取出来进行傅里叶变换得到 V 通道频率信号;接着使用巴特沃斯滤波器对频域信号进行滤波得到 V 通道高频信号;下一步将 V 通道高频信号和原 V 通道频率信号进行叠加用以增强纹理细节;然后对增强的 V 通道频率信号进行傅里叶反变换并进行直方图均衡化滤波实现对图像增强局部的对比度而不影响整体的对比,从而提高图像整体亮度和对比度;最后将处理后得到的 V 通道信号替换原始图像 V 通道信号,并输出增强图像。这样则实现了眼科图像的纹理增强,且避免了过曝与失真问题。

二、算法原理和步骤

1. 获取原图像,并对图像进行颜色空间转换

由于如果直接对原始图像的灰度图进行处理,则最后得到的也是灰度增强的图像,而无法体现在彩色图像上;此外,如果对 RGB 通道分别进行滤波增强,则最后得到的图像会导致颜色失真,与要求不符。故在为了能对亮度通道进行单独处理,需要将原图像转换为 HSV 颜色模式。

2. 获取亮度 V 通道信号进行傅里叶变换

对于 HSV 颜色空间中, H(Hue,色相)和 S(Saturation,饱和度)的处理也会导致颜色失真等问题,所以需要提取亮度 V 通道信号,并对其做傅里叶变换。

3. 对频域 V 通道进行巴特沃斯滤波,得到高/低通频谱

在图像处理中,巴特沃斯滤波器在抑制噪声的同时还能减小边缘模糊,并且没有振铃效应, n 阶截至频率为 D_0 的巴特沃斯低通滤波器 (BLPF) 传递函数为:

$$H(u, v) = \frac{1}{1 + [D(u, v) / D_0]^{2n}} \quad (1)$$

其中 $D(u, v) = \sqrt{u^2 + v^2}$ 。对于一幅输入的图像 $f(x, y)$, 其可以分解为高频分量 $f_h(x, y)$ 和低频分量 $f_l(x, y)$, 当图像通过巴特沃斯低通滤波器时, 获得图像的低频分量为:

$$f_l(x, y) = f(x, y) * H(u, v) \quad (2)$$

那么对应图像中的高频分量为：

$$f_h(x,y)=f(x,y)-f_l(x,y) \tag{3}$$

4. 将频域高频信号叠加在原亮度信号，并做傅里叶反变换和直方图均衡化


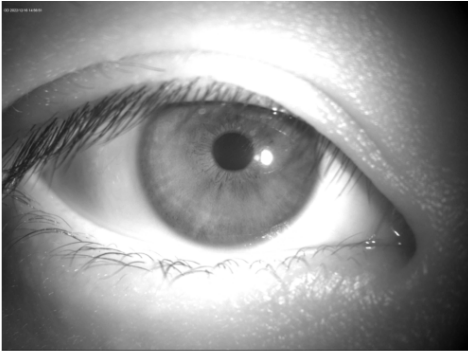
由于高频信号代表了图像的纹理和细节，这样将其叠加在原 V 通道信号上实现图像增强。接着通过对 V 通道信号做傅里叶反变换，在时域下进行 V 通道信号直方图均衡化进一步增强全局图像亮度和局部对比度。

5. 将最终图像增强的 V 通道信号替换至原图中，并将最终图像进行保存

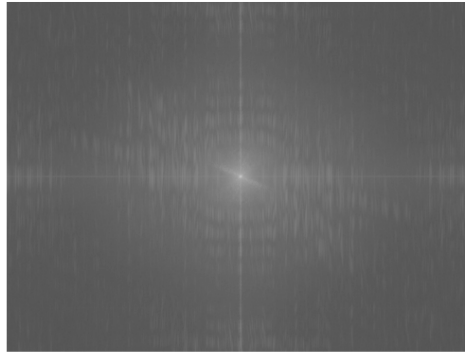
这样即在不改变颜色或导致过曝的情况下实现了眼科图像纹理增强并保存下了增强后的图像。

三、实验结果分析

一）图像增强结果及分析

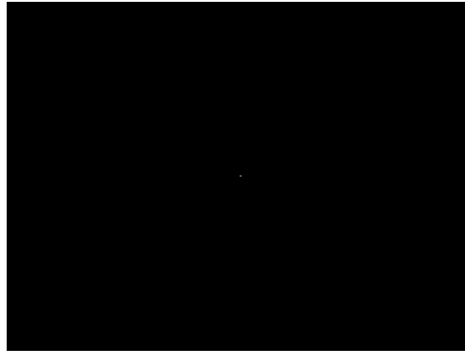
1.首先获取并显示原始图像。
<div>raw_imgRGB (原始图像)</div> 
2.接着通过颜色模型转换获取 V 通道的图像信号。
<div>img_V (图像V通道时域图)</div> 
3.对上述图像做傅里叶变换后得到的频谱图如下图所示。

fimg(图像V通道频域图)



4.对 V 通道频域信号进行巴特沃斯滤波，由于根据 V 通道信号的频域集中在 1~12，在本实验中选择截至频率为 5.7 的 4 阶巴特沃斯滤波，用以更好的分离高频信号。

巴特沃斯低通滤波器



5.通过傅里叶反变换，得到的高频和低频信号如下，以明显看出高频信号所体现的图像纹理细节，以及低频图像描述的为图像的主要部分。

h_filtered_img(滤波后图像V通道高频时域图)



l_filtered_img (滤波后图像V通道低频域图)



6.将高频 V 通道信号叠加在原 V 通道上，通过傅里叶反变换看到的图像如下。

img_V_R (叠加增强的V通道时域图)

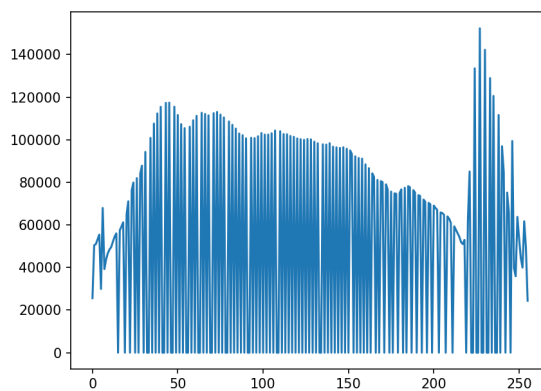
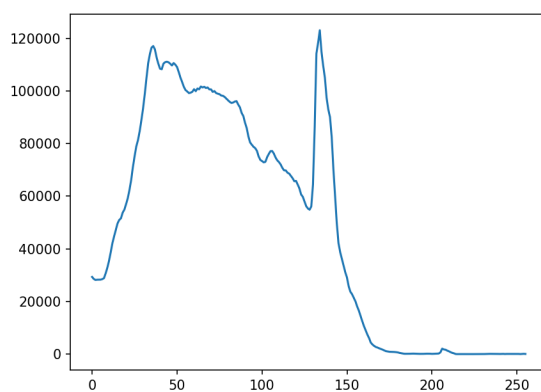


7.此时如果直接将 V 通道的信号进行替换，得到增强的图像如下，可以看出经过图像的纹理和细节得到了增强但是图像的整体亮度降低了，不适合人眼观察。

result_RGB (最终图像增强效果图)



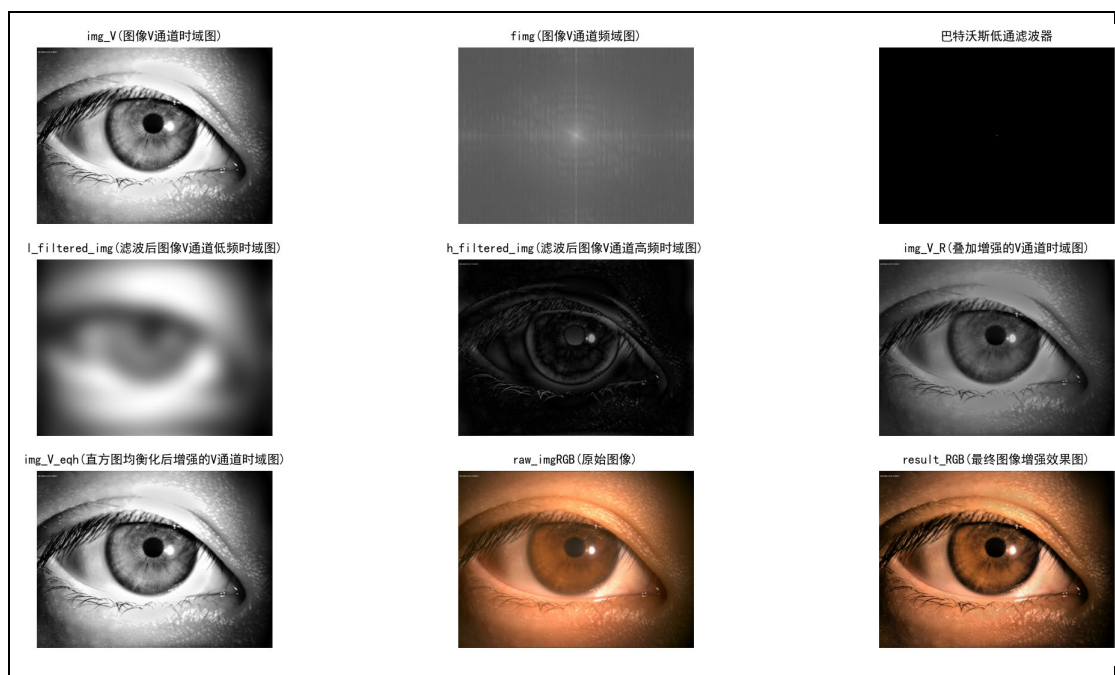
8.此时通过观察增强 V 通道信号的灰度直方图可以看出其灰度值主要集中在 150 以下的位置，可以通过直方图均衡化扩展灰度范围，提高全局亮度。



9.最后得到增强输出的眼科图像如下图。



10.图像增强过程图如下所示。



二) 算法耗时分析

首先不同步骤对应的算法说明如下：

步骤	算法说明
一	获取原图像，并对图像进行颜色空间转换
二	获取亮度 V 通道信号，并进行傅里叶变换
三	对频域亮度值进行巴特沃斯滤波，进一步得到高/低通频谱
四	将频域高频信号叠加在原亮度信号，并做傅里叶反变换和直方图均衡化
五	将最终图像增强的 V 通道信号替换至原图中，并将最终图像进行保存

通过 5 次运行整个算法仿真实验，得到的各步骤与总体耗时如下表(时间单位为:s)

次数 步骤	一	二	三	四	五	总耗时
1	0.213704	0.544669	30.475109	0.970983	0.519082	32.723547
2	0.208473	0.653424	30.645532	0.918656	0.521350	32.947435
3	0.210044	0.552125	30.727108	0.975400	0.528111	32.992788
4	0.210235	0.541014	30.810560	0.948346	0.510665	33.02082
5	0.213037	0.536381	30.765394	0.922887	0.513704	32.951403
平均耗时	0.2110986	0.5655226	30.6847406	0.9472544	0.5185824	32.9271986

从上表可以看出，整个算法的平均耗时为：32.9271986 秒，算法处理较为的复杂繁琐的地方为步骤三，由于图像的像素为 4056X3040，图像的像素比较多导致在产生与像素一一对应巴特沃斯滤波数值时候需要消耗较多的时间。

由于通过显示循环产生对应长宽的巴特沃斯滤波器，使得算法时间较长，在此使用 NumPy 的向量化操作，而不是显式循环，以加速滤波器的计算。NumPy 提供了许多数学操作，如幂运算和开方运算，可以使用这些函数来提高计算效率。

改进后的通过 5 次运行整个算法，得到结果如下：

次数 步骤	一	二	三	四	五	总耗时
-------	---	---	---	---	---	-----

1	0.212905	0.542669	0.727101	0.985051	0.547551	3.015277
2	0.214813	0.543725	0.730190	0.931032	0.515510	2.93527
3	0.223691	0.540012	0.739184	0.946141	0.524013	2.973041
4	0.219647	0.540576	0.734408	0.920149	0.508756	2.923536
5	0.208280	0.615744	0.727945	0.925264	0.527468	3.004701
平均耗时	0.2158672	0.5565452	0.7317656	0.9415274	0.5246596	2.970365

可以明显看出利用 NumPy 库的向量计算功能后，步骤三的时间下降了几近 40 倍，整体算法运行时间缩短了 30 倍之多。

改进前后的代码对比如下：

改进前的巴特沃斯滤波器
<pre>def butterworth_filter (img, order, cutoff_frequency): """ butterworth filter genreator $H(u, v) = 1 / (1 + (D(u, v) / cutoff_frequency)^{(2 * order)})$ Args: img: 输入灰度图 order: 滤波器阶数 cutoff_frequency: 截至频率 """ # 中心位置 h, w = img.shape[0],img.shape[1] cx, cy = w // 2, h // 2 # 优化代码前 tmp = np.zeros((h, w)) for i in range(h): for j in range(w): dis= np.sqrt((i - cy) ** 2 + (j - cx) ** 2) tmp[i, j] = 1 / (1 + (dis / cutoff_frequency) ** (2 * order)) return tmp</pre>
改进后的巴特沃斯滤波器
<pre>def butterworth_filter (img, order, cutoff_frequency): """ butterworth filter genreator $H(u, v) = 1 / (1 + (D(u, v) / cutoff_frequency)^{(2 * order)})$ Args: img: 输入灰度图 order: 滤波器阶数 cutoff_frequency: 截至频率 """ # 中心位置 h, w = img.shape[0],img.shape[1] cx, cy = w // 2, h // 2 # 优化代码后 # 创建频率域网格</pre>


```
u, v = np.meshgrid(np.arange(w) - cx, np.arange(h) - cy)
D = np.sqrt(u ** 2 + v ** 2)
# 计算滤波器
tmp = 1 / (1 + (D / cutoff_frequency) ** (2 * order))
return tmp
```

三) 算法性能分析

增强方法一为本实验的增强方法外；方法二为前文提及对叠加高频增强后不做直方图均衡化得到的结果；方法三为另一种增强方法：首先对图像灰度图进行直方图均衡化，自适应阈值化和对比度调整最后纹理增强处理，并将其与原始图像合并为彩色 RGB 图像。将上述三种方法分别简称为：Poposed_method, N_eqh_method 和 Another_method，并将上述三种图像的 RGB 图进行对比。

Another_method	N_eqh_method	Poposed_method
		

根据上图对比可以看出 Another_method 存在过曝光的情况，并且纹理增强较弱，出现类似给出的差评示例中的情况；_eqh_method 纹理细节得到的大幅度提高，但是其亮度对比度较低，不适于人眼观察；Poposed_method 方法实现了纹理和对比度提高，亮度增强都较好的效果。

此外，进一步通过图像质量评价指标峰值信噪比（PSNR）和结构相似性指数（SSIM）进行评价。其中 PSNR 是一种用于评估图像质量的指标，通常用于比较原始图像和经过压缩或其他处理后的图像之间的相似性。PSNR 值越高，表示图像质量越好。SSIM 是一种用于图像质量评估的指标，它衡量了两个图像之间的结构相似性。SSIM 值的范围通常在-1 到 1 之间，1 表示两个图像非常相似，-1 表示两个图像差异非常大。通常，较高的 SSIM 值表示更好的图像质量。对比度则是得到图像的亮度差异。通过计算得到三种方案的 PSNR、SSIM 指标和对比度值如下：

	Another_method	N_eqh_method	Poposed_method
PSNR	18.219184	12.750801	23.488828
SSIM	0.944306	0.681444	0.879311
对比度	54.093338	14.755621	55.239245

从上表可知，对于峰值信噪比指标本实验提出的方法的更优的；对于结构相似性指数指标 Another_method 的数值较好，但是实际其因为通过被处理的图像与原图按照比例叠加导致其数值更好，而是增强效果均不如本文提出的方法。对于对比度指标，本文提出的方法多得到的增强图像也具有更大的数值。

综上所述，本文所使用的方法可以较好的增强眼科图像的纹理细节，得到了较好的保真度和对比度，防止颜色失真和过曝光等问题。