**河南科技学院2023-2024学年第二学期**

**《数字图像处理应用》**

**系统设计说明书**

**简单图像处理操作集合系统设计**

**学 院 名 称： 人 工 智 能 学 院**

**专 业 名 称： 人工智能**

**年 级 班 别： 223班**

**小 组 成 员 （学号，姓名）：**

**20222214303 党硕硕**

**20222214301 程景晗**

**20222214302 程文豪**

**20222214305 惠子卿**

**任 课 教 师： 郑建锋**

**2024年06月**

**目录**

[一、系统需求分析 3](#_Toc170394708)

[二、系统设计说明 3](#_Toc170394709)

[1系统总体设计 3](#_Toc170394710)

[2开发方法 3](#_Toc170394711)

[3前端设计 6](#_Toc170394712)

[3.1基本组件 6](#_Toc170394713)

[3.2图像处理功能选项卡 6](#_Toc170394714)

[3.3布局 7](#_Toc170394715)

[三、关键程序代码 7](#_Toc170394716)

[1自动调整 8](#_Toc170394717)

[1.1降噪 8](#_Toc170394718)

[1.2伽马校正 8](#_Toc170394719)

[1.3自动白平衡 8](#_Toc170394720)

[1.4自动对比度增强 9](#_Toc170394721)

[2图像几何变换 10](#_Toc170394722)

[2.1裁剪 10](#_Toc170394723)

[2.2旋转 11](#_Toc170394724)

[2.3翻转 11](#_Toc170394725)

[3图像调节 11](#_Toc170394726)

[3.1亮度、饱和度、色调调节 11](#_Toc170394727)

[4图像输入和输出 12](#_Toc170394728)

[4.1用户上传图像部分 12](#_Toc170394729)

[4.2图像输出部分 13](#_Toc170394730)

[4.3图像处理部分 13](#_Toc170394731)

[四、小组分工说明 15](#_Toc170394732)

[参考资料 16](#_Toc170394733)

# 系统需求分析

这个图像处理APP旨在满足广泛的用户需求，包括个人摄影爱好者、社交媒体内容创作者、设计师、电子商务、教育、科研、图像恢复和广告市场营销等多个领域。针对个人摄影爱好者，APP提供简单易用的图像编辑工具，如亮度、对比度、饱和度调整以及基本的裁剪和旋转功能。社交媒体内容创作者可以利用APP快速优化和美化图片与视频，提升内容的吸引力和互动性。

设计师和艺术家需要高级图像处理功能，如降噪、伽马校正和自动白平衡，满足他们的专业创作需求。电子商务用户则通过自动调整、背景移除等功能优化产品图片，提高销售转化率。教育和科研领域用户可以使用APP处理教学和实验图像，增强展示效果。图像恢复和档案数字化方面，APP提供降噪、色彩校正和划痕修复等功能，帮助恢复和保存历史图像资料。广告和市场营销人员则可利用APP创作和优化广告图像，提高视觉吸引力和宣传效果。

# 系统设计说明

## 1系统总体设计

总体来说，APP主要实现以下几个部分。其中最关键的部分是处理图像部分,可以将其转化为各种基本处理函数的组合调用。

界面设计，初步计划如下：

本次开发APP使用的开发环境是MATLAB R2020b,这个版本APP Designer有了许多有用的更新，比如uitable也加入了列宽定义为可变宽度，按钮和面板组加入了Enable选项等实用的功能。

## 2开发方法

APP Designer使用[面向对象语言编程](https://ww2.mathworks.cn/products/matlab/object-oriented-programming.html)，页面分为设计视图和代码视图界面，实现了前端和后端的分离，可以分别进行设计，最后组合在一起。 下图是设计视图，左侧是一些组件，这些组件本身有一些属性可以在右侧查看，如位置，字体颜色，交互性等。而每个组件的回调函数，可以让用户的行为触发相应的函数，响应用户操作。

App程序采用面向对象设计模式，声明对象、定义函数、设置属性和共享数据都封装在一个类中，一个MLAPP文件就是一个类的定义。

**① 类的基本结构** properties段：属性的定义，主要包含属性声明代码。 methods段：方法的定义，由若干函数组成。回调函数只有一个参数app，为界面句柄，存储了界面中各个成员的数据。

classdef 类名 < matlab.apps.AppBase   
   properties (Access = public)   
   …   
   end   
   methods (Access = private)   
       function 函数1(app)   
       …   
       end   
       function 函数2(app)   
       …   
       end   
   end   
end

存取数据和调用函数称为访问对象成员。

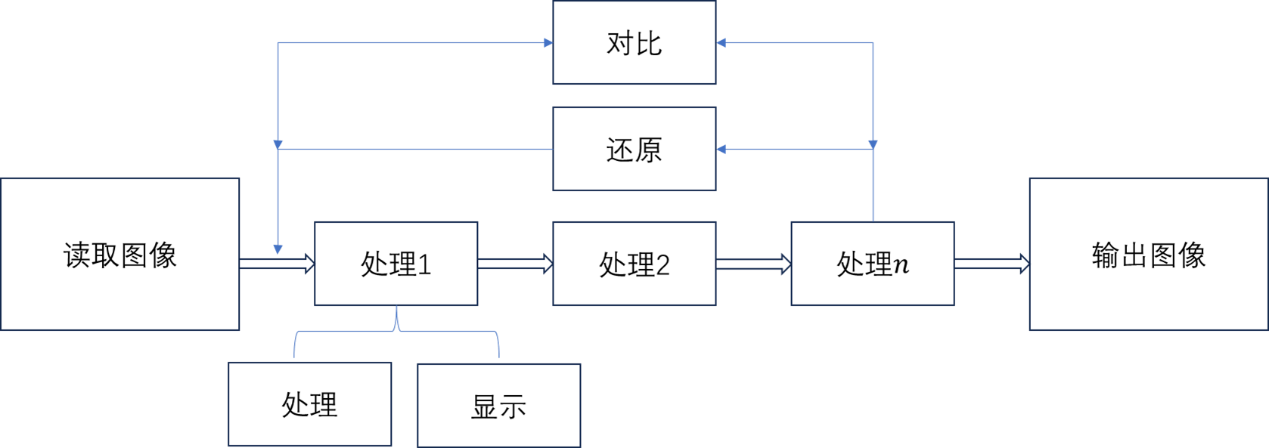
**② 访问权限** public：可用于与App的其他类共享数据。 private：只允许在本类中访问。 以下属性和函数的默认访问权限为private。

* 属性的声明
* 界面的启动函数startupFcn
* 建立界面组件的函数createComponents
* 回调函数

下面是空白APP的代码，可以清晰的看清代码的结构。

classdef app1 < matlab.apps.AppBase  
​  
   % Properties that correspond to app components  
   properties (Access = public)  
       UIFigure  matlab.ui.Figure  
   end  
​  
   % Component initialization  
   methods (Access = private)  
​  
       % Create UIFigure and components  
       function createComponents(app)  
​  
           % Create UIFigure and hide until all components are created  
           app.UIFigure = uifigure('Visible', 'off');  
           app.UIFigure.Position = [100 100 640 480];  
           app.UIFigure.Name = 'MATLAB App';  
​  
           % Show the figure after all components are created  
           app.UIFigure.Visible = 'on';  
       end  
   end  
​  
   % App creation and deletion  
   methods (Access = public)  
​  
       % Construct app  
       function app = app1  
​  
           % Create UIFigure and components  
           createComponents(app)  
​  
           % Register the app with App Designer  
           registerApp(app, app.UIFigure)  
​  
           if nargout == 0  
               clear app  
           end  
       end  
​  
       % Code that executes before app deletion  
       function delete(app)  
​  
           % Delete UIFigure when app is deleted  
           delete(app.UIFigure)  
       end  
   end  
end

本次项目代码的整体结构如下



## 3前端设计

## 3.1基本组件

UIFigure：应用程序的主窗口。

GridLayout：网格布局，用于管理其他组件的布局。

LoadButton：按钮，用于导入图像。

Button\_8：状态按钮（StateButton），用于对比操作。

Button\_9：按钮，用于复原图像到最初状态。

Button\_11：按钮，用于导出处理后的图像。

TabGroup：选项卡组，包含多个选项卡用于不同的图像处理功能。

图像展示与调节相关组件

ImageAxes：用于展示图像的轴（UIAxes）。

RedAxes, GreenAxes, BlueAxes：分别用于展示红色、绿色、蓝色通道的直方图的轴（UIAxes）。

## 3.2图像处理功能选项卡

Tab\_1：图像处理的第一个选项卡，包含自动白平衡、降噪、自动对比度增强和伽马矫正的按钮。

Button\_10（自动白平衡按钮）

Button\_5（降噪按钮）

Button\_6（自动对比度增强按钮）

Button\_13（伽马矫正按钮）

Tab\_2：图形调节的选项卡，包含裁剪、旋转、水平镜像和垂直镜像的按钮。

Button\_2（裁剪按钮）

Button\_3（旋转按钮）

Button\_4（水平镜像按钮）

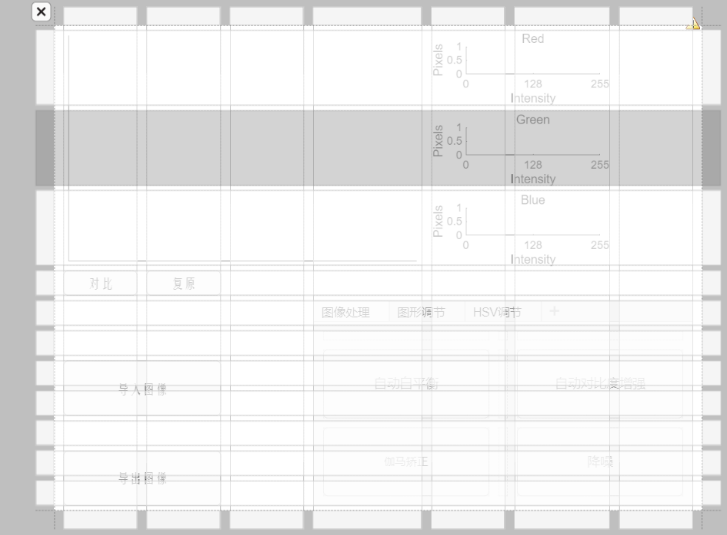
Button\_12（垂直镜像按钮）

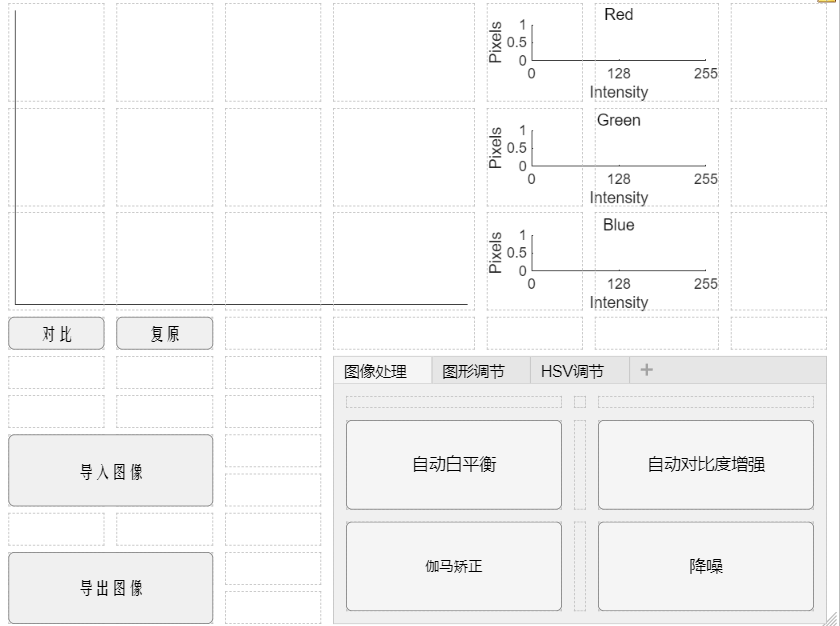
HSVTab：HSV调节的选项卡，包含亮度、饱和度和色调的滑块（Slider）和标签（Label）。

Slider\_1, Slider\_2, Slider\_3（分别对应亮度、饱和度、色调的调节滑块）

Label\_2, Label\_3, Label\_4（对应的标签，分别标注“亮度”、“饱和度”、“色调”）

## 3.3布局

使用网格布局，配置加权大小。并开启图窗的位置中的Resize属性，使其能根据窗口大小自动调整布局。接下来拖动相应的组件，并编辑。

设置好相应的回调函数之后，打开APP，最终效果如下

# 关键程序代码

## 1自动调整

## 1.1降噪

[wiener2](https://ww2.mathworks.cn/help/images/ref/wiener2.html?s_tid=doc_ta):二维自适应去噪滤波 基于图像的局部统计特性来估计噪声方差，并根据噪声的特性进行滤波。这种滤波方法通常在存在噪声的图像中能够有效地减少噪声并保持图像的细节。

function denoise(app, event)  % 去噪按钮按下时执行的函数  
   if app.im\_class == 3  
       J(:,:,1) = wiener2(app.im(:,:,1),[3 3]);  
       J(:,:,2) = wiener2(app.im(:,:,2),[3 3]);  
       J(:,:,3) = wiener2(app.im(:,:,3),[3 3]);  
       updateimage(app,J);  
   else  
       J = wiener2(app.im,[3 3]);  
       updateimage(app,J);  
   end  
end

## 1.2伽马校正

[lin2rgb](https://ww2.mathworks.cn/help/images/ref/lin2rgb.html?s_tid=doc_ta):将线性 RGB 值应用伽马校正，使其转换为适合显示的 sRGB 色彩空间。对图像中的像素值进行非线性变换，使较暗区域的细节更加可见，同时保持较亮区域的细节不被过度压缩。这样可以增强图像的对比度，使其在显示时更加生动和自然。

function gamma(app, event)  % gamma按钮按下时执行的函数  
   if (app.im\_class == 3)  
       sRGB = lin2rgb(app.im); % 将线性RGB图像转换为sRGB图像  
       updateimage(app,sRGB);  
         
   end  
end

## 1.3自动白平衡

当人们用眼晴观察自然世界时，在不同的光线下，对相同颜色的感觉基本是相同的，大脑已经对不同光线下的物体的彩色还原有了适应性。这种现象称为颜色恒常性。不幸的是，CMOS或CCD等感光器件没有这样的适应能力。 为了使得摄像机也具有颜色恒常性能力，需要使用白平衡技术。所谓白平衡(WiteBalance),简单地说就是去除环境光的影响，还原物体真实的颜色，把不同色温下的白颜色调整正确。从理论上说白颜色调整正确了，其他色彩就都准确了。即在红色灯光照射下，白色物体依然呈白色，在蓝色灯光照射下也呈现白色。 灰度世界算法以灰度世界假设为基础，该假设认为：对于一幅有着大量色彩变化的图像,其R,G,B 三个色彩分量的平均值趋于同一灰度值 K。 从物理意义上讲，灰色世界法假设自然界景物对于光线的平均反射的均值在总体上是个定值，这个定值近似地为“灰色”。 颜色平衡算法将这一假设强制应用于待处理图像，可以从图像中消除环境光的影响，获得原始场景图像。

function AWB(app, event)   % 白平衡  
   R = app.im(:,:,1);G = app.im(:,:,2);B = app.im(:,:,3);  
   Rave = mean2(R);  
   Gave = mean2(G);  
   Bave = mean2(B);  
   K = (Rave + Gave + Bave) / 3;  
     
   R\_new=(K/Rave)\*R;G\_new=(K/Gave)\*G;B\_new=(K/Bave)\*B;  
   J = (cat(3,R\_new,G\_new,B\_new));  
   updateimage(app,J);  
end

## 1.4自动对比度增强

MATLAB中有三个函数适用于对比度增强：

* imadjust：将输入强度图像的值映射到新值，以对输入数据中强度最低和最高的 1%（默认值）数据进行饱和处理，从而提高图像的对比度。
* histeq：执行直方图均衡化。它变换强度图像中的值，以使输出图像的直方图近似匹配指定的直方图（默认情况下为均匀分布），从而增强图像的对比度。
* adapthisteq ：执行对比度受限的自适应直方图均衡化。与 histeq 不同，它对小数据区域（图块）而不是整个图像执行运算。它会增强每个图块的对比度，使得每个输出区域的直方图近似匹配指定的直方图（默认情况下为均匀分布）。可以限制对比度增强，以避免放大图像中可能存在的噪声。 通过对比以下各图，可以看出adapthisteq方法效果最好。 [adapthisteq](https://ww2.mathworks.cn/help/images/ref/adapthisteq.html?s_tid=doc_ta):对比度受限的自适应直方图均衡化 (CLAHE) 通过在LAB空间中对亮度通道进行处理，我们能够保持彩色图像的整体色彩平衡，因为在LAB空间中，色度通道（a和b通道）与亮度通道（L通道）是相互独立的。这种方法可以在增强对比度的同时保持图像的自然色彩。

function CLAHE(app, event)  
       if (app.im\_class == 3)  
           lab\_image = rgb2lab(app.im);  
           % 提取亮度通道,将值缩放到 adapthisteq 函数预期的范围 [0 1]。  
           luminance = lab\_image(:, :, 1)/100;  
           % 对亮度通道进行自适应直方图均衡化,缩放结果，使其回到 L\*a\*b\* 颜色空间使用的范围。  
           enhanced\_luminance = adapthisteq(luminance)\*100;  
           % 替换原来的亮度通道  
           lab\_image(:,:,1) = enhanced\_luminance;  
           % 将图像转换回RGB空间  
           enhanced\_image = lab2rgb(lab\_image);  
           updateimage(app,enhanced\_image);  
       else  
           J = adapthisteq(app.im);  % 灰度图像直方图均衡化增强对比度  
           updateimage(app,J);  
       end  
end

## 2图像几何变换

## 2.1裁剪

[imcrop](https://ww2.mathworks.cn/help/images/ref/imcrop.html?s_tid=doc_ta)：imcrop 创建与当前图窗中显示的灰度、真彩色或二值图像相关联的交互式裁剪图像工具。imcrop 返回裁剪的图像 Icropped。

      function crop(app, event)   % 裁剪按钮按下时执行的函数  
           J = imcrop(app.im);  
           if ~isempty(J)       % 如果用户没有取消裁剪  
               updateimage(app,J);  
               close all  
           end  
       end

## 2.2旋转

[imrotate](https://ww2.mathworks.cn/help/images/ref/imrotate.html?s_tid=doc_ta):J = imrotate(I,angle)将图像 I 围绕其中心点逆时针方向旋转 angle 度。要顺时针旋转图像，请为 angle 指定负值。imrotate 使输出图像 J 足够大，可以包含整个旋转图像。默认情况下，imrotate 使用最近邻点插值，对于数值和逻辑图像，将 J 中位于旋转后的图像外的像素的值设置为 0；对于分类图像，设置为 missing。

function rotation(app, event)  % 旋转按钮按下时执行的函数  
   J = imrotate(app.im,-90,'bilinear','crop'); % 逆时针旋转90度  
   updateimage(app,J);  
end

## 2.3翻转

[fliplr](https://ww2.mathworks.cn/help/matlab/ref/fliplr.html?s_tid=doc_ta):围绕垂直轴按左右方向翻转其各列。对于多维数组，fliplr 作用于由第一个和第二个维度构成的平面。 [flipud](https://ww2.mathworks.cn/help/matlab/ref/flipud.html?s_tid=doc_ta):围绕水平轴按上下方向翻转其各行。

function flip1(app, event)  % 水平翻转按钮按下时执行的函数  
   J = fliplr(app.im);      
   updateimage(app,J);  
end  
   function flip2(app, event)   % 垂直翻转图像  
   J = flipud(app.im);  
   updateimage(app,J);  
end

## 3图像调节

## 3.1亮度、饱和度、色调调节

HSV模型的三个分量（色相、饱和度、亮度）是相互独立的，调节其中一个分量不会直接影响其他分量。 [rgb2hsv](https://ww2.mathworks.cn/help/matlab/ref/rgb2hsv.html?s_tid=doc_ta):将 RGB 图像的红色、绿色和蓝色值转换为 HSV 图像的色调、饱和度和明度 (HSV) 值。 [hsv2rgb](https://ww2.mathworks.cn/help/matlab/ref/hsv2rgb.html?s_tid=doc_ta):将 HSV 图像的色调、饱和度和明度值转换为 RGB 图像的红色、绿色和蓝色值。

function chge(app, event)  % 滑动条改变时执行的函数  
   if (app.im\_class == 3)  
       H = 0.002\*app.Slider\_3.Value+1; % 色调调节因子  
       S = 0.01\*app.Slider\_2.Value+1;  % 饱和度调节因子  
       V = 0.01\*app.Slider\_1.Value+1;  % 亮度调节因子  
       hsv = app.im\_hsv;  
       hsv(:,:,1) = hsv(:,:,1)\*H; % 色调  
       hsv(:,:,2) = hsv(:,:,2)\*S;% 饱和度  
       hsv(:,:,3) = hsv(:,:,3)\*V; % 亮度  
       J = hsv2rgb(hsv);  
       updateimage(app,J);  
   else  
       app.im = app.im\_tab\*(1+app.Slider\_1.Value/100);  
       % 灰度图像亮度调节  
       updateimage(app,app.im);  
   end  
end

## 4图像输入和输出

## 4.1用户上传图像部分

function LoadButtonPushed(app, event)  
   
 % Display uigetfile dialog  
 filterspec = {'\*.jpg;\*.tif;\*.png;\*.gif','All Image Files'};  
 [f, p] = uigetfile(filterspec);  
   
 % Make sure user didn't cancel uigetfile dialog  
 if (ischar(p))  
 fname = [p f];  
 updateimage(app, fname);  
 app.im\_init = app.im;  
 reset(app,0)  
 end  
end

这部分主要用到了uigetfile这个内置函数，该函数可以打开文件选择对话框，让用户选择相应文件，并返回文件的名称和路径，同样上传后不做处理直接输出。

## 4.2图像输出部分

这部分比较简单，直接调用内置的imsava函数即可。

function export(app, event)  
 imsave(app.ImageAxes);  
end

## 4.3图像处理部分

输入和输出的功能已经实现，图像处理作为一个整体，又可分解为一次次基本图像处理的组合。如下图所示，每次处理之后为了直观展现处理效果，方便调节，所以也要进行显示。而还原是和对比又为每次处理提供了更多的选择性。

图像处理部分由于分解为按步处理，所以要定义属性，方便不同处理函数之间共享数据。且对比和还原功能也需要相应的属性。

properties (Access = public)  
 im % 当前显示的图像数据  
 im\_hsv % 打开调节1选项卡时保存的HSV空间数据  
 im\_class % 图像类别数据  
 im\_tab % 选项卡图像数据  
 im\_init % 初始图像数据  
end

如果对当前图像进行某个处理，便可以将im的属性传递给该处理函数，处理之后，调用图像显示函数，更新im并显示图像。 将im\_init传入图像更新函数，便可将im恢复到im\_init，实现**图像复原**。 由于图像更新后im属性也会改变，**图像对比**要定义一个局部变量暂存现在的图像。

function compare(app, event)  
 if app.Button\_8.Value == 1  
 im\_0 = app.im;  
 updateimage(app,app.im\_init);  
 app.im = im\_0;  
 else  
 updateimage(app,app.im);  
 end  
end

不同于图像处理部分需要根据处理方式设计不同的函数，图像显示可以设计为可复用函数的形式，将某图像和其直方图显示出来。 由于为输入图像传入或者为处理后图像传入，为了实现函数的复用，所以进行判断选择，而且加入错误提示。 图像更新函数updateimage(app,imagefile)分解如下：

try  
 if (ischar(imagefile))  
 app.im = imread(imagefile);  
 else  
 app.im = imagefile;  
 end  
   
catch ME  
 % If problem reading image, display error message  
 uialert(app.UIFigure, ME.message, 'Image Error');  
 return;  
end

之后是图像的显示输出。这里根据图像类型为灰度还是真彩色，分别进行不同的显示。

app.im\_class = size(app.im,3);  
if isempty(app.im\_tab)  
 app.im\_tab = app.im;  
 app.im\_init = app.im;  
end  
% Create histograms based on number of color channels  
switch app.im\_class  
 case 1  
 % Display the grayscale image  
 imagesc(app.ImageAxes,app.im);  
   
 % Plot all histograms with the same data for grayscale  
 histr = histogram(app.RedAxes, app.im, 'FaceColor',[1 0 0],'EdgeColor', 'none');  
 histg = histogram(app.GreenAxes, app.im, 'FaceColor',[0 1 0],'EdgeColor', 'none');  
 histb = histogram(app.BlueAxes, app.im, 'FaceColor',[0 0 1],'EdgeColor', 'none');  
   
   
 case 3  
   
 % Display the truecolor image  
 imagesc(app.ImageAxes,app.im);  
   
 % Plot the histograms  
 histr = histogram(app.RedAxes, im2uint8(app.im(:,:,1)), 'FaceColor', [1 0 0], 'EdgeColor', 'none');  
 histg = histogram(app.GreenAxes, im2uint8(app.im(:,:,2)), 'FaceColor', [0 1 0], 'EdgeColor', 'none');  
 histb = histogram(app.BlueAxes, im2uint8(app.im(:,:,3)), 'FaceColor', [0 0 1], 'EdgeColor', 'none');  
   
 otherwise  
 % Error when image is not grayscale or truecolor  
 uialert(app.UIFigure, 'Image must be grayscale or truecolor.', 'Image Error');  
 return;  
end  
% Get largest bin count  
maxr = max(histr.BinCounts);  
maxg = max(histg.BinCounts);  
maxb = max(histb.BinCounts);  
maxcount = max([maxr maxg maxb]);  
  
% Set y axes limits based on largest bin count  
app.RedAxes.YLim = [0 maxcount];  
app.RedAxes.YTick = round([0 maxcount/2 maxcount], 2, 'significant');  
app.GreenAxes.YLim = [0 maxcount];  
app.GreenAxes.YTick = round([0 maxcount/2 maxcount], 2, 'significant');  
app.BlueAxes.YLim = [0 maxcount];  
app.BlueAxes.YTick = round([0 maxcount/2 maxcount], 2, 'significant');

**四、小组分工说明**

|  |  |
| --- | --- |
| 成员 | 负责模块 |
| 党硕硕 | 后端代码和模块功能的整合 |
| 程文豪 | 前端界面的设计 |
| 惠子卿、程景晗 | 需求分析和说明书的编写 |

**参考资料**

1. [Image Processing Toolbox Documentation - MathWorks 中国](https://ww2.mathworks.cn/help/images/?s_tid=srchbrcm)
2. [对比度增强方法 - MATLAB & Simulink - MathWorks 中国](https://ww2.mathworks.cn/help/images/contrast-enhancement-techniques.html)
3. [App 构建组件 - MATLAB & Simulink - MathWorks 中国](https://ww2.mathworks.cn/help/matlab/creating_guis/choose-components-for-your-app-designer-app.html)