PROJECT GUI DESIGN PROCEDURE

Bold = Code/Component used

Red Font = App Component Names

Green Font = GUI Displayed Titles/Texts

1) Open App Designer (>>**appdesigner)**

2) Discuss Functionality
   a) Group/Row 1
      i) Finite Impulse Response (FIR) Filter (Different Types)
      ii) Original minus Background Image
      iii) Any additional GUI (Image Brightness)

   b) Group/Row 2
      i) Binary/Bilevel Image (Image Type)
      ii) RGB/Truecolor Image (Image Type)
      iii) Intensity/Gracscale Image (Image Type)
      iv) Morphological Opening to Estimate the Background

   c) Group/Row 3
      i) Indexed/Pseudocolor Image (Image Type)
      ii) Multiframe Image
      iii) Image Contrast Adjustment
      iv) Image Type of Original Image (Text output)

3) Include Appropriate Components in the Design View
   a) UIFigure
      i) Position = 0,0,1024,1024
      ii) WindowState = maximized
      iii) Resize (yes)

   b) GUI Labels (**Label**) (font 18, bold)
      i) Project Title (Top, Center)

(1) MATLAB Image Processing Project with GUI, "app.ProjectTitle"

c) GUI Group Sections (**Panel**)
   i) Filter Characteristics (Top-left Panel), "app.FilterCharacteristicsPanel"
   ii) Image Selection (Bottom-left Panel), "app.ImageSelectionPanel"
   iii) Filtered Images (Right Panel), "app.FilteredImagesPanel"

d) Filtered Images Panel (3x1 Grid), "app.GridLayout1"
   i) Group/Row 1 Filtered Image, "app.R1_Filtered_Image"
   ii) Group/Row 2 Filtered Image, "app.R2_Filtered_Image"
   iii) Group/Row 3 Filtered Image, "app.R3_Filtered_Image"

e) Image Selection Panel
   i) Original Image Sample, "app.Original_Image"
   ii) Select Image, "app.SelectImageButton"
   iii) Reset, "app.ResetButton"

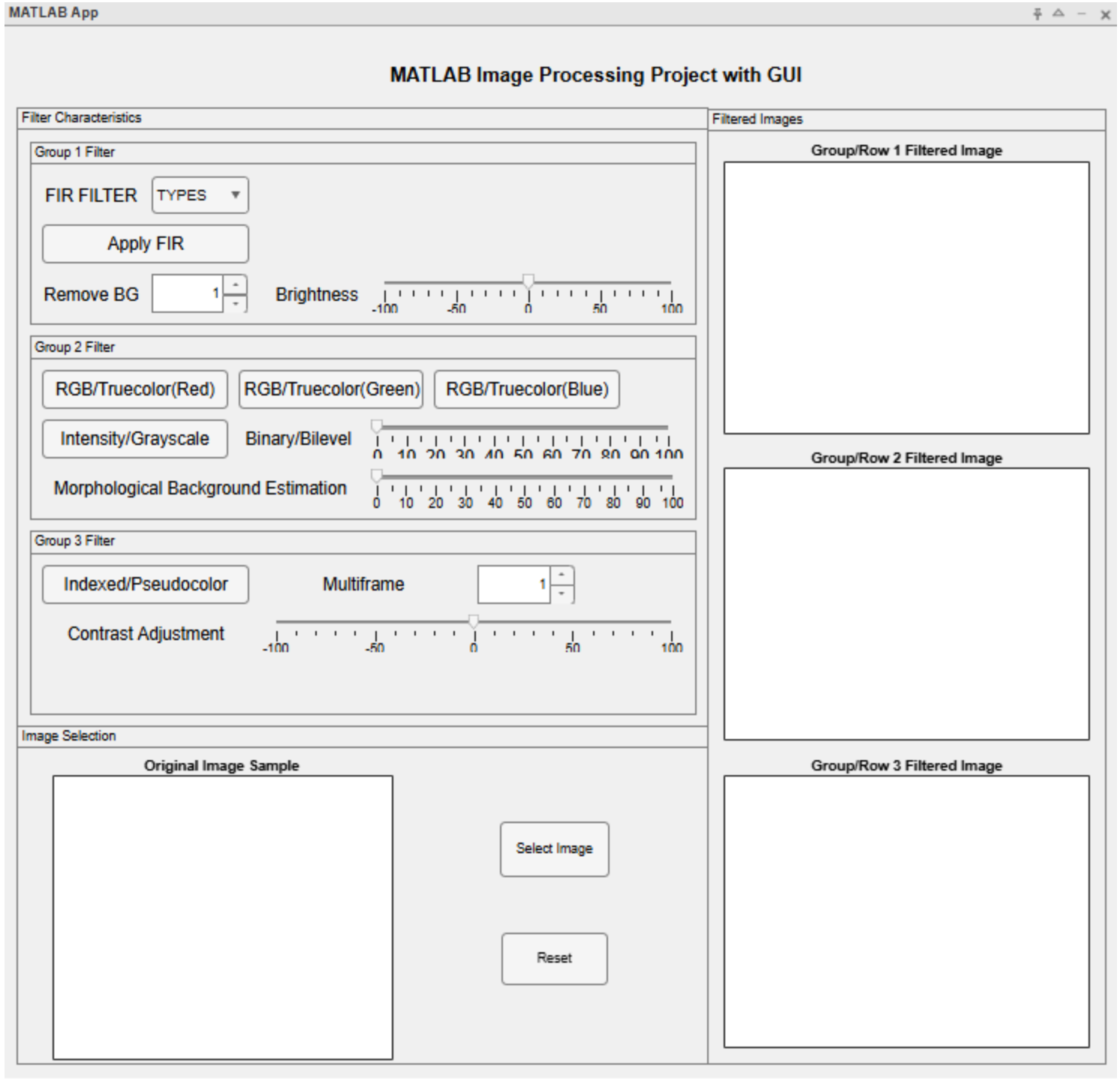
f) Filter Characteristics Panel (**3x1 Grid**), "app.GridLayout2"
   i) Group 1 Filter Panel (**3x6 Grid**), "app.Group1FilterPanel", "app.GridLayout3"
   ii) Group 2 Filter Panel (**3x10 Grid**), "app.Group2FilterPanel", "app.GridLayout4"
   iii) Group 3 Filter Panel (**3x6 Grid**), "app.Group3FilterPanel", "app.GridLayout5"

   iv) Group/Row 1 Filter
      (1) FIR FILTER (**Drop Down**), , "app.FIRFilter", (font 16), (font 14 for cases)
         (a) TYPES
         (b) Low Pass Filter, LPF
         (c) High Pass Filter, HPF
         (d) Band Pass Filter, BPF
         (e) Band Stop Filter, BSF
      (2) Lower Cut-off Frequency (**Label**), "app.LowerCutoffLabel", (1 grid), (font 14), (Not Visible)
      (3) Lower Cut-off Frequency (**Slider**), "app.LowerCutoffSlider", (3 grids), (range from 0 – 100), (Not Visible)
      (4) Upper Cut-off Frequency (**Label**), "app.UpperCutoffLabel",(1 grid), (font 14)
      (5) Upper Cut-off Frequency (**Slider**), "app.UpperCutoffSlider" , (3 grids), (range from 0 – 100), (Not Visible)
      (6) Apply FIR (**Button**), "app.ApplyFIRButton", (2 grids), (font 16), (Not Visible)
      (7) Original minus Background Image (**Spinner**), Remove BG, "app.RemoveBGSpinner", (font 16), (2 grids), (value 1)
      (8) Image Brightness (**Slider**), "app.BrightnessSlider", (1,3 grids), (range from -100 – 100), (font 16)

v) Group/Row 2 Filter
   (1) RGB/Truecolor(Red) (**Button**), "app.RGBRedButton", (3 grids), (font 16)
   (2) RGB/Truecolor(Green) (**Button**), "app.RGBGreenButton", (3 grids), (font 16)
   (3) RGB/Truecolor(Bluee) (**Button**), "app.RGBBlueButton", (3 grids), (font 16)
   (4) Intensity/Grayscale Image Type (**Button**), "app.IntensityGrayscaleButton", (3 grids), (font 16)
   (5) Binary/Bilevel Image Type(**Slider**), "app.BinaryBilevelSlider", (2,5 grids), (range from 0 – 100), (font 16)
   (6) Morphological Background Estimation (**Slider**), "app.BgEstimationSlider", (5,5 grids), (range from 0 – 100), (font 16)

vi) Group/Row 3 Filter
   (1) Indexed/Pseudocolor Image Type (**Button**), "app.IndexedPseudocolorButton", (2 grids), (font 16)
   (2) Multiframe (**Spinner)**, "app.MultiframeSpinner" , (2,1 grids), (font 16), (value start at 1)
   (3) Image Contrast Adjustment (**Slider**), "app.ContrastAdjustmentSlider" , (2,4 grids), (font 16), (range from -100 – 100)

EXPECTED GUI OUTPUT DESIGN


MATLAB App
MATLAB Image Processing Project with GUI
Filter Characteristics
Group 1 Filter
FIR FILTER
TYPES
Apply FIR
Remove BG
1
Brightness
-100
-50
0
50
100
Group 2 Filter
RGB/Truecolor(Red)
RGB/Truecolor(Green)
RGB/Truecolor(Blue)
Intensity/Grayscale
Binary/Bilevel
0 10 20 30 40 50 60 70 80 90 100
Morphological Background Estimation
0 10 20 30 40 50 60 70 80 90 100
Group 3 Filter
Indexed/Pseudocolor
Multiframe
1
Contrast Adjustment
-100
-50
0
50
100
Image Selection
Original Image Sample
Select Image
Reset
Filtered Images
Group/Row 1 Filtered Image
Group/Row 2 Filtered Image
Group/Row 3 Filtered Image

MATLAB App

MATLAB Image Processing Project with GUI

Filter Characteristics

Group 1 Filter

FIR FILTER TYPES

Apply FIR

Remove BG 1

Brightness -100 -50 0 50 100

Group 2 Filter

RGB/Truecolor(Red) RGB/Truecolor(Green) RGB/Truecolor(Blue)

Intensity/Grayscale

Binary/Bilevel 0 10 20 30 40 50 60 70 80 90 100

Morphological Background Estimation 0 10 20 30 40 50 60 70 80 90 100

Group 3 Filter

Indexed/Pseudocolor

Multiframe 1

Contrast Adjustment -100 -50 0 50 100

Image Selection

Original Image Sample

Select Image

Reset

Filtered Images

Group/Row 1 Filtered Image

Group/Row 2 Filtered Image

Group/Row 3 Filtered Image

PROJECT CODE PROCEDURE

Gray shade = Non-coding Segments

White shade = Coding Segments

Violet Font = Type of Callback

1) Create a Private Property to share Variables between functions
   a) App Designer >> Code view >> Property Tab >> **Private Property**

2) Add Callback functions to the components which will be used to filter
   a) Image Selection Panel
      i) Select Image Button, "SelectImageButtonPushed"
      ii) Reset Button, "ResetButtonPushed"

   b) Filter Characteristics Panel
      i) Filter 1
         (1) FIR Filter Drop Down, "FIRFILTERDropDownValueChanged"
         (2) Lower Cut-off Slider, "LowerCutoffSliderValueChanged"
         (3) Upper Cut-off Slider, "UpperCutoffSliderValueChanged"
         (4) Apply FIR Button, "ApplyFIRButtonPushed"
         (5) Remove Background Spinner, "RemoveBackgroundSpinnerValueChanged"
         (6) Brightness, "BrightnessSliderValueChanged"

      ii) Filter 2
         (1) RGB Red Button, "RGBRedButtonPushed"
         (2) RGB Green Button, "RGBGreenButtonPushed"
         (3) RGB Blue Button, "RGBBlueButtonPushed"
         (4) Intensity/Grayscale Button, "IntensityGrayscaleButtonPushed"
         (5) Binary Slider, "BinaryBilevelSliderValueChanged"
         (6) Background Estimation Slider, "MorphologicalBackgroundEstimationSliderValueChanged"

      iii) Filter 3
         (1) Indexed/Pseudocolor Button, "IndexedPseudocolorButtonPushed"
         (2) Multiframe Spinner, "MultiframeSpinnerValueChanged"
         (3) Contrast Adjustment Slider, "ContrastAdjustmentSliderValueChanged"

3) Add necessary function names for each callbacks

a) Button pushed function: SelectImageButton
   i) Image_Sample(app);

b) Value changed function: FIRFILTERDropDown
   i) FIR_Type_Select(app, app.FIRFILTERDropDown.Value);

c) Value changed function: LowerCutoffSlider
   i) app.LCF = app.LowerCutoffSlider.Value;

d) Value changed function: UpperCutoffSlider
   i) app.UCF = app.UpperCutoffSlider.Value;

e) Button pushed function: ApplyFIRButton
   i) switch app.FIRFILTERDropDown.Value
      (1)case 'TYPES'
         (a)FIR(app, app.FIRFILTERDropDown.Value, 0, 0);
      (2)case 'LPF'
         (a)FIR(app, app.FIRFILTERDropDown.Value, 0, app.UCF);
      (3)case 'HPF'
         (a)FIR(app, app.FIRFILTERDropDown.Value, app.LCF, 0);
      (4)case 'BPF'
         (a)FIR(app, app.FIRFILTERDropDown.Value, app.LCF, app.UCF);
      (5)case 'BSF'
         (a)FIR(app, app.FIRFILTERDropDown.Value, app.LCF, app.UCF);

f) Value changed function: RemoveBGSpinner
   i) BG_Remove(app, app.RemoveBGSpinner.Value);

g) Value changed function: BrightnessSlider
   i) Brightness(app, app.BrightnessSlider.Value);

h) Button pushed function: RGBRedButton
   i) RGB_Red(app);

i) Button pushed function: RGBGreenButton
   i) RGB_Green(app);

j) Button pushed function: RGBBlueButton
   i) RGB_Blue(app);

```
k) Button pushed function: IntensityGrayscaleButton
   i) Intensity(app);

l) Value changed function: BinaryBilevelSlider
   i) Binary(app, app.BinaryBilevelSlider.Value);

m) Value changed function: MorphologicalBackgroundEstimationSlider
   i) BG_Estimate(app, app.MorphologicalBackgroundEstimationSlider.Value);

n) Button pushed function: IndexedPseudocolorButton
   i) Index(app);

o) Value changed function: MultiframeSpinner
   i) Multiframe(app, app.MultiframeSpinner.Value);

p) Value changed function: ContrastAdjustmentSlider
   i) Contrast(app, app.ContrastAdjustmentSlider.Value);

q) Button pushed function: ResetButton
   i) Reset(app)
```

4) Add a **Private Property** for variables used in the program
   a) Selected_Images (image sample)
   b) Image_1 (image for group 1 filter)
   c) Image_2 (image for group 2 filter)
   d) Image_3 (image for group 3 filter)
   e) LCF (variable to contain the lower cut-off frequency)
   f) UCF (variable to contain the upper cut-off frequency)
   g) Edited_Image_1 (image to display in group 1 filter axes)
   h) Edited_Image_2 (image to display in group 2 filter axes)
   i) Edited_Image_3 (image to display in group 3 filter axes)

5) Add necessary functions to process/filter the image based on the component values by the user
   a) Selecting and Displaying a Sample Image file
      i) >>**help uigetfile**

```
function Image_Sample(app)
    [filename, pathname] = uigetfile('*.*', 'Pick a MATLAB code file');
    if isequal(filename,0) || isequal(pathname,0)
        disp('User pressed cancel')
    else
        filename = strcat(pathname, filename);
        app.Selected_Image = imread(filename);
        imshow(app.Selected_Image, 'Parent', app.Original_Image);
        app.Image_1 = app.Selected_Image;
        app.Image_2 = app.Selected_Image;
        app.Image_3 = app.Selected_Image;
    end
end
```

   b) Selecting Filter Type
      i) >>**help switch**

```
function FIR_Type_Select(app, type)
    switch type
        case 'TYPES'
            app.LowerCutoffSlider.Visible = 'off';
            app.LowerCutoffLabel.Visible = 'off';
            app.UpperCutoffSlider.Visible = 'off';
            app.UpperCutoffLabel.Visible = 'off';

        case 'LPF'
            app.LowerCutoffSlider.Visible = 'off';
            app.LowerCutoffLabel.Visible = 'off';
            app.UpperCutoffSlider.Visible = 'on';
            app.UpperCutoffLabel.Visible = 'on';

        case 'HPF'
            app.LowerCutoffSlider.Visible = 'on';
            app.LowerCutoffLabel.Visible = 'on';
            app.UpperCutoffSlider.Visible = 'off';
            app.UpperCutoffLabel.Visible = 'off';
```

```
            case 'BPF'
                app.LowerCutoffSlider.Visible = 'on';
                app.LowerCutoffLabel.Visible = 'on';
                app.UpperCutoffSlider.Visible = 'on';
                app.UpperCutoffLabel.Visible = 'on';

            case 'BSF'
                app.LowerCutoffSlider.Visible = 'on';
                app.LowerCutoffLabel.Visible = 'on';
                app.UpperCutoffSlider.Visible = 'on';
                app.UpperCutoffLabel.Visible = 'on';
        end
    end
```

c) FIR Filtering (LPF, HPF, BPF, BSF)

i) >>**help switches**

(1) LPF

(a) >>**help imgaussfilt**

(2) HPF

(a) >>**help imsharpen**

(3) BPF

(a) External source code, **gaussianbpf**

(4) BSF

(a) >>**help rgb2ind**

(b) >>**help medfilt2**

```
function FIR(app, type, LFc, HFc)
    switch type
        case 'TYPES'
            imshow(app.Image_1, 'Parent', app.R1_Filtered_Image);

        case 'LPF'
            if HFc ~= 0
                edit = imgaussfilt(app.Selected_Image, 10.01 - (HFc / 10));
                imshow(edit, 'Parent', app.R1_Filtered_Image);
                app.Edited_Image_1 = edit;
            else
                imshow(app.Image_1, 'Parent', app.R1_Filtered_Image);
            end
```

```
        case 'HPF'
            if LFc ~= 0
                edit = imsharpen(app.Selected_Image,'Amount', LFc / 10);
                imshow(edit, 'Parent', app.R1_Filtered_Image);
                app.Edited_Image_1 = edit;
            else
                imshow(app.Image_1, 'Parent', app.R1_Filtered_Image);
            end

        case 'BPF'
            edit = gaussianbpf(app, app.Selected_Image,LFc,HFc);
            [R, C] = size(edit);
            crop = imcrop(edit, [0, 0, R, C]);
            imshow(crop, 'Parent', app.R1_Filtered_Image);
            app.Edited_Image_1 = crop;


        case 'BSF'
            [edit, map] = rgb2ind(app.Selected_Image,2);
            edit = medfilt2(edit, [round(LFc/10) round(HFc/10)]);
            imshow(edit, map, 'Parent', app.R1_Filtered_Image);
            app.Edited_Image_1 = edit;
    end
end


function filtered_image = gaussianbpf(~, I,d0,d1)
    f = double(I);
    [nx ny] = size(f);
    f = uint8(f);
    fftI = fft2(f,2*nx-1,2*ny-1);
    fftI = fftshift(fftI);

    % Initialize filter.
    filter1 = ones(2*nx-1,2*ny-1);
    filter2 = ones(2*nx-1,2*ny-1);
    filter3 = ones(2*nx-1,2*ny-1);


    for i = 1:2*nx-1
        for j =1:2*ny-1
            dist = ((i-(nx+1))^2 + (j-(ny+1))^2)^.5;
```

```
            % Use Gaussian filter.
            filter1(i,j) = exp(-dist^2/(2*d1^2));
            filter2(i,j) = exp(-dist^2/(2*d0^2));
            filter3(i,j) = 1.0 - filter2(i,j);
            filter3(i,j) = filter1(i,j).*filter3(i,j);
        end
    end

    % Update image with passed frequencies
    filtered_image = fftI + filter3.*fftI;
    filtered_image = ifftshift(filtered_image);
    filtered_image = ifft2(filtered_image,2*nx-1,2*ny-1);
    filtered_image = real(filtered_image(1:nx,1:ny));
    filtered_image = uint8(filtered_image);
end
```

d) Removing Background

i) >>**help imopen**

ii) >>**help strel**

iii) >>**help imsubtract**

```
function BG_Remove(app, radius)
    background = imopen(app.Selected_Image, strel('disk', radius,  4));
    BG_Removed = imsubtract(app.Selected_Image, background);
    imshow(BG_Removed, 'Parent', app.R1_Filtered_Image);
    app.Edited_Image_1 = BG_Removed;
end
```

e) Adjusting Image Brightness

```
function Brightness(app, brightness)
    if brightness ~= 0
        edit = app.Image_1 + round(brightness);
        imagesc(app.R1_Filtered_Image, edit);
        axis(app.R1_Filtered_Image,'image');
        app.Edited_Image_1 = app.Image_1;
    else
        imagesc(app.R1_Filtered_Image, app.Image_1);
        axis(app.R1_Filtered_Image,'image');
```

```
    end
end
```

f) RGB/Truecolor Red Image

```
function RGB_Red(app)
    edit = app.Image_2;
    edit(:,:,2:3) = 0;
    imagesc(app.R2_Filtered_Image, edit);
    axis(app.R2_Filtered_Image,'image');
    app.Edited_Image_2 = edit;
end
```

g) RGB/Truecolor Green Image

```
function RGB_Green(app)
    edit = app.Image_2;
    edit(:,:,1) = 0;
    edit(:,:,3) = 0;
    imagesc(app.R2_Filtered_Image, edit);
    axis(app.R2_Filtered_Image,'image');
    app.Edited_Image_2 = edit;
end
```

h) RGB/Truecolor Blue Image

```
function RGB_Blue(app)
    edit = app.Image_2;
    edit(:,:,1:2) = 0;
    imagesc(app.R2_Filtered_Image, edit);
    axis(app.R2_Filtered_Image,'image');
    app.Edited_Image_2 = edit;
end
```

i) Intensity/Grayscale Image

i) >>**help im2gray**

```
function Intensity(app)
```

```
        edit = im2gray(app.Image_2);
        imshow(edit, 'Parent',app.R2_Filtered_Image);
        axis(app.R2_Filtered_Image,'image');
        app.Edited_Image_2 = edit;
    end
```

j) Binary/Bilevel Image
   i) >>**im2bw**

```
    function Binary(app, binary)
        if binary ~= 0
            edit = im2bw(app.Image_2, binary/100); %#ok<IM2BW>
            imshow(edit, 'Parent', app.R2_Filtered_Image);
            axis(app.R2_Filtered_Image,'image');
            app.Edited_Image_2 = edit;
        else
            imagesc(app.R2_Filtered_Image, app.Image_2);
            axis(app.R2_Filtered_Image,'image');
        end
    end
```

k) Background Estimation
   i) >>**help imopen**
   ii) >>**help strel**

```
    function BG_Estimate(app, estimated)
        if estimated ~= 0
            edit = imopen(app.Image_2, strel('disk', round(estimated), 4));
            imagesc(app.R2_Filtered_Image, edit);
            axis(app.R2_Filtered_Image,'image');
            app.Edited_Image_2 = edit;
        else
            imagesc(app.R2_Filtered_Image, app.Image_2);
            axis(app.R2_Filtered_Image,'image');
        end
    end
```

l) Indexed/Pseudocolor Image
   i) >>**help rgb2ind**

ii) >>**help colormap**

```
function Index(app)
    [ind, map] = rgb2ind(app.Image_3, 8);
    imagesc(app.R3_Filtered_Image, ind);
    axis(app.R3_Filtered_Image,'image');
    colormap(app.R3_Filtered_Image,map);
    app.Edited_Image_3 = app.Image_3;
end
```

m) Multriframing

i) >>**help repmat**

```
function Multiframe(app, multiframe)
    edit = repmat(app.Image_3, [multiframe multiframe]);
    imagesc(app.R3_Filtered_Image, edit);
    axis(app.R3_Filtered_Image,'image');
    app.Edited_Image_3;
end
```

n) Adjusting Image Contrast

i) >>**help localcontrast**

```
function Contrast(app, contrast)
    if contrast ~= 0
        edit = localcontrast(app.Image_3, 0.3, contrast/100);
        imagesc(app.R3_Filtered_Image, edit);
        axis(app.R3_Filtered_Image,'image');
        app.Edited_Image_3 = app.Image_3;
    else
        imagesc(app.R3_Filtered_Image, app.Image_3);
        axis(app.R3_Filtered_Image,'image');
    end
end
```

o) Reset Program Changes

i) Overwrite edited images with the sample image

ii) Clear axes

(1) >>**help cla**

iii) Rewrite Component Values with the Initial Values

```
function Reset(app)
    %Reset all filtered images
    app.Image_1 = app.Selected_Image;
    app.Image_2 = app.Selected_Image;
    app.Image_3 = app.Selected_Image;
    app.Edited_Image_1 = app.Selected_Image;
    app.Edited_Image_2 = app.Selected_Image;
    app.Edited_Image_3 = app.Selected_Image;

    %Reset filtered displays
    cla(app.R1_Filtered_Image);
    cla(app.R2_Filtered_Image);
    cla(app.R3_Filtered_Image);

    %Reset all user input data to the initial values
    %Group 1 Filter
    app.FIRFILTERDropDown.Value = 'TYPES';
    app.LCF = 0;
    app.LowerCutoffLabel.Visible = 'off';
    app.LowerCutoffSlider.Visible = 'off';
    app.LowerCutoffSlider.Value = 0;
    app.UCF = 0;
    app.UpperCutoffLabel.Visible = 'off';
    app.UpperCutoffSlider.Visible = 'off';
    app.UpperCutoffSlider.Value = 0;
    app.RemoveBGSpinner.Value = 1;
    app.BrightnessSlider.Value = 0;

    %Group 2 Filter
    app.BinaryBilevelSlider.Value = 0;
    app.MorphologicalBackgroundEstimationSlider.Value = 0;

    %Group 3 Filter
    app.MultiframeSpinner.Value = 1;
    app.ContrastAdjustmentSlider.Value = 0;
end
```