```
classdef app1 < matlab.apps.AppBase
    % Properties that correspond to app components
    properties (Access = public)
        PhotoEditor                 matlab.ui.Figure
        UIAxes                      matlab.ui.control.UIAxes
        SelectImageButton           matlab.ui.control.Button
        SaveButton                  matlab.ui.control.Button
        TabGroup                    matlab.ui.container.TabGroup
        ResizeTab                   matlab.ui.container.Tab
        TabGroup2                   matlab.ui.container.TabGroup
        CropTab                     matlab.ui.container.Tab
        SelectCropButton            matlab.ui.control.Button
        TextArea                    matlab.ui.control.TextArea
        RotateTab                   matlab.ui.container.Tab
        FreerotateKnob              matlab.ui.control.Knob
        FlipHorizontalSwitchLabel   matlab.ui.control.Label
        FlipHorizontalSwitch        matlab.ui.control.Switch
        FlipVerticalSwitchLabel     matlab.ui.control.Label
        FlipVerticalSwitch          matlab.ui.control.Switch
        ApplyRotateButton           matlab.ui.control.Button
        ColorAdjustTab              matlab.ui.container.Tab
        TabGroup3                   matlab.ui.container.TabGroup
        BrightnessTab               matlab.ui.container.Tab
        ApplyBrightnessButton       matlab.ui.control.StateButton
        BrightnessSlider            matlab.ui.control.Slider
        ContrastTab                 matlab.ui.container.Tab
        ContrastSlider              matlab.ui.control.Slider
        ApplyContrastButton         matlab.ui.control.StateButton
        SaturationTab               matlab.ui.container.Tab
        SaturationSlider            matlab.ui.control.Slider
        ApplySaturationButton       matlab.ui.control.StateButton
        TemperatureTab              matlab.ui.container.Tab
        TemperatureSlider           matlab.ui.control.Slider
        ApplyTemperatureButton      matlab.ui.control.StateButton
        SharpnessTab                matlab.ui.container.Tab
        SharpnessSlider             matlab.ui.control.Slider
        ApplySharpnessButton        matlab.ui.control.StateButton
        HistogramTab                matlab.ui.container.Tab
        HistogramAxes               matlab.ui.control.UIAxes
        AutoAdjustButton            matlab.ui.control.Button
        OverlayTab                  matlab.ui.container.Tab
        FiltersListBoxLabel         matlab.ui.control.Label
        FiltersListBox              matlab.ui.control.ListBox
        ApplyFilterButton           matlab.ui.control.Button
        HistoryTab                  matlab.ui.container.Tab
        ListBox                     matlab.ui.control.ListBox
        RollBackButton              matlab.ui.control.Button
    end
    properties (Access = private)
        imagefile;% variable for the image to be stored in
        img;
        orgimg;
        croppedimg;
        rotateimg;
        brightimg;
```

```
    contrastimg;
    saturationimg;
    sharpimg;
    temperatureimg;
    filterimg;
    autoimg;
    histimg;
end
methods (Access = private)
    % Button pushed function: SelectImageButton
    function SelectImageButtonPushed(app, event)
        app.imagefile = uigetfile('*.jpg','Select a File');
        imshow(app.imagefile,'Parent',app.UIAxes);
        app.img=imread(app.imagefile);
        app.orgimg=app.img;
        histogram(app.HistogramAxes,app.img);
    end
    % Button pushed function: SaveButton
    function SaveButtonPushed(app, event)
        imwrite(app.img,uiputfile({'*.png'; '*.jpg'}));
    end
    % Button pushed function: SelectCropButton
    function SelectCropButtonPushed(app, event)
        [cropimg,positions] =  imcrop(app.img);
        imshow(cropimg,'Parent',app.UIAxes);
        app.croppedimg=cropimg;
        app.img=cropimg;
        histogram(app.HistogramAxes,app.img);
    end
    % Value changed function: FreerotateKnob
    function FreerotateKnobValueChanged(app, event)
        value = app.FreerotateKnob.Value;
        app.rotateimg = imrotate(app.img, value );
        imshow(app.rotateimg,'Parent',app.UIAxes);
    end
    % Value changed function: FlipHorizontalSwitch
    function FlipHorizontalSwitchValueChanged(app, event)
        value = app.FlipHorizontalSwitch.Value;
        switch value
            case 'on'
                app.img = flip(app.img,2);
               imshow(app.img,'Parent',app.UIAxes);
            case 'off'
                app.img = flip(app.img,2);
               imshow(app.img,'Parent',app.UIAxes);
        end
        histogram(app.HistogramAxes,app.img);
    end
    % Value changed function: FlipVerticalSwitch
    function FlipVerticalSwitchValueChanged(app, event)
        value = app.FlipVerticalSwitch.Value;
        switch value
            case 'on'
                app.img = flip(app.img,1);
               imshow(app.img,'Parent',app.UIAxes);
```

```
            case 'off'
                app.img = flip(app.img,1);
               imshow(app.img,'Parent',app.UIAxes);
        end
        histogram(app.HistogramAxes,app.img);
    end
    % Callback function
    function DoneCheckBoxValueChanged(app, event)
        value = app.DoneCheckBox.Value;
        app.img=app.brightimg;
        histogram(app.HistogramAxes,app.img);
    end
    % Value changed function: FiltersListBox
    function FiltersListBoxValueChanged(app, event)
        value = app.FiltersListBox.Value;
        switch value
            case 'None'
                imshow(app.img,'Parent',app.UIAxes);
            case 'Black & White'
                grayimg=rgb2gray(app.img);
                app.filterimg=grayimg;
               imshow(app.filterimg,'Parent',app.UIAxes);
            case 'Sepia'
                inputRed = app.img(:,:,1); %// Extract each colour plane
                inputGreen = app.img(:,:,2);
                inputBlue = app.img(:,:,3);
                %// Create sepia tones for each channel
                outputRed = (inputRed * .393) + (inputGreen *.769) + (inputBlue * .189);
                outputGreen = (inputRed * .349) + (inputGreen *.686) + (inputBlue *
.168);
                outputBlue = (inputRed * .272) + (inputGreen *.534) + (inputBlue * .131);
                %// Create output image by putting all of these back into a 3D matrix
                %// and convert back to uint8
                sepiaimg = uint8(cat(3, outputRed, outputGreen, outputBlue));
                %sepiaimg=ind2rgb(grayimg,pink);
                app.filterimg=sepiaimg;
                imshow(app.filterimg,'Parent',app.UIAxes);
            case 'Negative'
                negimg=imcomplement(app.img);
                app.filterimg=negimg;
                imshow(app.filterimg,'Parent',app.UIAxes);
            case 'Winter'
                inputRed = app.img(:,:,1); %// Extract each colour plane
                inputGreen = app.img(:,:,2);
                inputBlue = app.img(:,:,3);
                outputRed = (inputRed * 0.237) + (inputGreen *.437) + (inputBlue * .237);
                outputGreen = (inputRed * .268) + (inputGreen *.421) + (inputBlue *
.307);
                outputBlue = (inputRed * .213) + (inputGreen *.253) + (inputBlue * .739);
                winterimg = uint8(cat(3, outputRed, outputGreen, outputBlue));
                app.filterimg=winterimg;
                imshow(app.filterimg,'Parent',app.UIAxes);
        end
    end
    % Button pushed function: ApplyFilterButton
```

```
function ApplyFilterButtonPushed(app, event)
    app.img=app.filterimg;
    histogram(app.HistogramAxes,app.img);
end
% Button pushed function: ApplyRotateButton
function ApplyRotateButtonPushed(app, event)
    app.img=app.rotateimg;
    histogram(app.HistogramAxes,app.img);
end
% Button pushed function: AutoAdjustButton
function AutoAdjustButtonPushed(app, event)
    adjust=histeq(app.img);
    app.autoimg=adjust;
    app.img=adjust;
    imshow(app.img,'Parent',app.UIAxes);
    histogram(app.HistogramAxes,app.img);
end
% Value changed function: BrightnessSlider
function BrightnessSliderValueChanged(app, event)
    value = app.BrightnessSlider.Value;
    app.brightimg=(app.img).*value;
    imshow(app.brightimg,'Parent',app.UIAxes);
end
% Value changed function: ContrastSlider
function ContrastSliderValueChanged(app, event)
    value = app.ContrastSlider.Value;
    app.contrastimg=app.img-value;
    imshow(app.contrastimg,'Parent',app.UIAxes);
end
% Value changed function: SaturationSlider
function SaturationSliderValueChanged(app, event)
    value = app.SaturationSlider.Value;
    HSV = rgb2hsv(app.img);
    HSV(:, :, 2) = HSV(:, :, 2) * value;
    HSV(HSV > 1) = 1;  % Limit values
    app.saturationimg = hsv2rgb(HSV);
    imshow(app.saturationimg,'Parent',app.UIAxes);
end
% Value changed function: SharpnessSlider
function SharpnessSliderValueChanged(app, event)
    value = app.SharpnessSlider.Value;
    app.sharpimg = imsharpen(app.img,'Radius',3,'Amount',value);
    imshow(app.sharpimg,'Parent',app.UIAxes);
end
% Value changed function: ApplySharpnessButton
function ApplySharpnessButtonValueChanged(app, event)
    value = app.ApplySharpnessButton.Value;
    app.img= app.sharpimg;
    histogram(app.HistogramAxes,app.img);
end
% Value changed function: ApplyBrightnessButton
function ApplyBrightnessButtonValueChanged(app, event)
    value = app.ApplyBrightnessButton.Value;
    app.img=app.brightimg;
    histogram(app.HistogramAxes,app.img);
```

```
        end
        % Value changed function: TemperatureSlider
        function TemperatureSliderValueChanged(app, event)
            value = app.TemperatureSlider.Value;
            if (value>=1)
                v=255-value;
                app.temperatureimg = chromadapt(app.img,uint8([1*v 2*v
3*v]),'ColorSpace','linear-rgb');
                imshow(app.temperatureimg,'Parent',app.UIAxes);
            elseif (value<= -1)
                v=256-(-1*value);
                app.temperatureimg = chromadapt(app.img,uint8([2*v 1*v
1*v]),'ColorSpace','linear-rgb');
                imshow(app.temperatureimg,'Parent',app.UIAxes);
            else
                app.temperatureimg=app.img;
                imshow(app.temperatureimg,'Parent',app.UIAxes);
            end
        end
        % Value changed function: ApplyTemperatureButton
        function ApplyTemperatureButtonValueChanged(app, event)
            value = app.ApplyTemperatureButton.Value;
            app.img=app.temperatureimg;
            histogram(app.HistogramAxes,app.img);
        end
        % Value changed function: ListBox
        function ListBoxValueChanged(app, event)
            value = app.ListBox.Value;
            switch value
                case 'Original Image'
                    app.histimg=app.orgimg;
                    imshow(app.histimg,'Parent',app.UIAxes);
                    histogram(app.HistogramAxes,app.histimg);
                case 'Cropped Image'
                    app.histimg=app.croppedimg;
                    imshow(app.histimg,'Parent',app.UIAxes);
                    histogram(app.HistogramAxes,app.histimg);
                case 'Rotated Image'
                    app.histimg=app.rotateimg;
                    imshow(app.histimg,'Parent',app.UIAxes);
                    histogram(app.HistogramAxes,app.histimg);
                case 'Brightness'
                    app.histimg=app.brightimg;
                    imshow(app.histimg,'Parent',app.UIAxes);
                    histogram(app.HistogramAxes,app.histimg);
                case 'Contrast'
                    app.histimg=app.contrastimg;
                    imshow(app.histimg,'Parent',app.UIAxes);
                    histogram(app.HistogramAxes,app.histimg);
                case 'Saturation'
                    app.histimg=app.saturationimg;
                    imshow(app.histimg,'Parent',app.UIAxes);
                    histogram(app.HistogramAxes,app.histimg);
                case 'Temperature'
                    app.histimg=app.temperatureimg;
```

```
                    imshow(app.histimg,'Parent',app.UIAxes);
                    histogram(app.HistogramAxes,app.histimg);
                case 'Sharpness'
                    app.histimg=app.sharpimg;
                    imshow(app.histimg,'Parent',app.UIAxes);
                    histogram(app.HistogramAxes,app.histimg);
                case 'Filter Applied Image'
                    app.histimg=app.filterimg;
                    imshow(app.histimg,'Parent',app.UIAxes);
                    histogram(app.HistogramAxes,app.histimg);
                case 'Auto Adjusted Image'
                    app.histimg=app.autoimg;
                    imshow(app.histimg,'Parent',app.UIAxes);
                    histogram(app.HistogramAxes,app.histimg);
            end
        end
        % Button pushed function: RollBackButton
        function RollBackButtonPushed(app, event)
            app.img=app.histimg;
        end
    end
    % App initialization and construction
    methods (Access = private)
        % Create UIFigure and components
        function createComponents(app)
            % Create PhotoEditor
            app.PhotoEditor = uifigure;
            app.PhotoEditor.Color = [0.149 0.149 0.149];
            app.PhotoEditor.Position = [100 100 945 583];
            app.PhotoEditor.Name = 'Photo Editor';
            % Create UIAxes
            app.UIAxes = uiaxes(app.PhotoEditor);
            title(app.UIAxes, 'Preview')
            app.UIAxes.Box = 'on';
            app.UIAxes.BackgroundColor = [0.8 0.8 0.8];
            app.UIAxes.Position = [340 34 587 529];
            % Create SelectImageButton
            app.SelectImageButton = uibutton(app.PhotoEditor, 'push');
            app.SelectImageButton.ButtonPushedFcn = createCallbackFcn(app,
@SelectImageButtonPushed, true);
            app.SelectImageButton.BackgroundColor = [0.502 0.502 0.502];
            app.SelectImageButton.FontColor = [1 1 1];
            app.SelectImageButton.Position = [120 541 100 22];
            app.SelectImageButton.Text = 'Select Image';
            % Create SaveButton
            app.SaveButton = uibutton(app.PhotoEditor, 'push');
            app.SaveButton.ButtonPushedFcn = createCallbackFcn(app, @SaveButtonPushed, true);
            app.SaveButton.BackgroundColor = [0.502 0.502 0.502];
            app.SaveButton.FontColor = [1 1 1];
            app.SaveButton.Position = [123 182 100 22];
            app.SaveButton.Text = 'Save';
            % Create TabGroup
            app.TabGroup = uitabgroup(app.PhotoEditor);
            app.TabGroup.Position = [16 232 314 286];
            % Create ResizeTab
```

```
            app.ResizeTab = uitab(app.TabGroup);
            app.ResizeTab.Title = 'Resize';
            % Create TabGroup2
            app.TabGroup2 = uitabgroup(app.ResizeTab);
            app.TabGroup2.TabLocation = 'left';
            app.TabGroup2.Position = [17 32 260 210];
            % Create CropTab
            app.CropTab = uitab(app.TabGroup2);
            app.CropTab.Title = 'Crop';
            % Create SelectCropButton
            app.SelectCropButton = uibutton(app.CropTab, 'push');
            app.SelectCropButton.ButtonPushedFcn = createCallbackFcn(app,
@SelectCropButtonPushed, true);
            app.SelectCropButton.Position = [46 167 100 22];
            app.SelectCropButton.Text = 'Select Crop';
            % Create TextArea
            app.TextArea = uitextarea(app.CropTab);
            app.TextArea.Position = [23 20 150 123];
            app.TextArea.Value = {'From the crop window:'; ''; '> Please select the crop area
and right click on it'; '> Choose the ''Crop Image'' option'; '> Close the crop window.'};
            % Create RotateTab
            app.RotateTab = uitab(app.TabGroup2);
            app.RotateTab.Title = 'Rotate';
            % Create FreerotateKnob
            app.FreerotateKnob = uiknob(app.RotateTab, 'continuous');
            app.FreerotateKnob.Limits = [0 360];
            app.FreerotateKnob.ValueChangedFcn = createCallbackFcn(app,
@FreerotateKnobValueChanged, true);
            app.FreerotateKnob.Position = [67 122 59 59];
            % Create FlipHorizontalSwitchLabel
            app.FlipHorizontalSwitchLabel = uilabel(app.RotateTab);
            app.FlipHorizontalSwitchLabel.HorizontalAlignment = 'center';
            app.FlipHorizontalSwitchLabel.Position = [13 44 82 22];
            app.FlipHorizontalSwitchLabel.Text = 'Flip Horizontal';
            % Create FlipHorizontalSwitch
            app.FlipHorizontalSwitch = uiswitch(app.RotateTab, 'slider');
            app.FlipHorizontalSwitch.Items = {'off', 'on'};
            app.FlipHorizontalSwitch.ValueChangedFcn = createCallbackFcn(app,
@FlipHorizontalSwitchValueChanged, true);
            app.FlipHorizontalSwitch.Position = [125 45 45 20];
            app.FlipHorizontalSwitch.Value = 'off';
            % Create FlipVerticalSwitchLabel
            app.FlipVerticalSwitchLabel = uilabel(app.RotateTab);
            app.FlipVerticalSwitchLabel.HorizontalAlignment = 'center';
            app.FlipVerticalSwitchLabel.Position = [13 15 68 22];
            app.FlipVerticalSwitchLabel.Text = 'Flip Vertical';
            % Create FlipVerticalSwitch
            app.FlipVerticalSwitch = uiswitch(app.RotateTab, 'slider');
            app.FlipVerticalSwitch.Items = {'off', 'on'};
            app.FlipVerticalSwitch.ValueChangedFcn = createCallbackFcn(app,
@FlipVerticalSwitchValueChanged, true);
            app.FlipVerticalSwitch.Position = [125 16 45 20];
            app.FlipVerticalSwitch.Value = 'off';
            % Create ApplyRotateButton
            app.ApplyRotateButton = uibutton(app.RotateTab, 'push');
```

```
            app.ApplyRotateButton.ButtonPushedFcn = createCallbackFcn(app,
@ApplyRotateButtonPushed, true);
            app.ApplyRotateButton.Position = [51 80 91 22];
            app.ApplyRotateButton.Text = 'Apply Rotate';
            % Create ColorAdjustTab
            app.ColorAdjustTab = uitab(app.TabGroup);
            app.ColorAdjustTab.Title = 'Color Adjust';
            % Create TabGroup3
            app.TabGroup3 = uitabgroup(app.ColorAdjustTab);
            app.TabGroup3.TabLocation = 'left';
            app.TabGroup3.Position = [13 21 290 200];
            % Create BrightnessTab
            app.BrightnessTab = uitab(app.TabGroup3);
            app.BrightnessTab.Title = 'Brightness';
            % Create ApplyBrightnessButton
            app.ApplyBrightnessButton = uibutton(app.BrightnessTab, 'state');
            app.ApplyBrightnessButton.ValueChangedFcn = createCallbackFcn(app,
@ApplyBrightnessButtonValueChanged, true);
            app.ApplyBrightnessButton.Text = 'Apply';
            app.ApplyBrightnessButton.Position = [47 124 100 22];
            % Create BrightnessSlider
            app.BrightnessSlider = uislider(app.BrightnessTab);
            app.BrightnessSlider.Limits = [0 2];
            app.BrightnessSlider.MajorTicks = 1;
            app.BrightnessSlider.MajorTickLabels = {'', '', '', '', '', ''};
            app.BrightnessSlider.ValueChangedFcn = createCallbackFcn(app,
@BrightnessSliderValueChanged, true);
            app.BrightnessSlider.MinorTicks = [];
            app.BrightnessSlider.Position = [14 173 166 3];
            app.BrightnessSlider.Value = 1;
            % Create ContrastTab
            app.ContrastTab = uitab(app.TabGroup3);
            app.ContrastTab.Title = 'Contrast';
            % Create ContrastSlider
            app.ContrastSlider = uislider(app.ContrastTab);
            app.ContrastSlider.Limits = [-50 50];
            app.ContrastSlider.MajorTicks = 0;
            app.ContrastSlider.MajorTickLabels = {};
            app.ContrastSlider.ValueChangedFcn = createCallbackFcn(app,
@ContrastSliderValueChanged, true);
            app.ContrastSlider.MinorTicks = [];
            app.ContrastSlider.Position = [14 173 166 3];
            % Create ApplyContrastButton
            app.ApplyContrastButton = uibutton(app.ContrastTab, 'state');
            app.ApplyContrastButton.Text = 'Apply';
            app.ApplyContrastButton.Position = [47 124 100 22];
            % Create SaturationTab
            app.SaturationTab = uitab(app.TabGroup3);
            app.SaturationTab.Title = 'Saturation';
            % Create SaturationSlider
            app.SaturationSlider = uislider(app.SaturationTab);
            app.SaturationSlider.Limits = [0 2];
            app.SaturationSlider.MajorTicks = 1;
            app.SaturationSlider.MajorTickLabels = {};
```

```
            app.SaturationSlider.ValueChangedFcn = createCallbackFcn(app,
@SaturationSliderValueChanged, true);
            app.SaturationSlider.MinorTicks = [];
            app.SaturationSlider.Position = [14 173 166 3];
            app.SaturationSlider.Value = 1;
            % Create ApplySaturationButton
            app.ApplySaturationButton = uibutton(app.SaturationTab, 'state');
            app.ApplySaturationButton.Text = 'Apply';
            app.ApplySaturationButton.Position = [47 124 100 22];
            % Create TemperatureTab
            app.TemperatureTab = uitab(app.TabGroup3);
            app.TemperatureTab.Title = 'Temperature';
            % Create TemperatureSlider
            app.TemperatureSlider = uislider(app.TemperatureTab);
            app.TemperatureSlider.Limits = [-255 255];
            app.TemperatureSlider.MajorTicks = 0;
            app.TemperatureSlider.MajorTickLabels = {};
            app.TemperatureSlider.ValueChangedFcn = createCallbackFcn(app,
@TemperatureSliderValueChanged, true);
            app.TemperatureSlider.MinorTicks = [];
            app.TemperatureSlider.Position = [14 173 166 3];
            % Create ApplyTemperatureButton
            app.ApplyTemperatureButton = uibutton(app.TemperatureTab, 'state');
            app.ApplyTemperatureButton.ValueChangedFcn = createCallbackFcn(app,
@ApplyTemperatureButtonValueChanged, true);
            app.ApplyTemperatureButton.Text = 'Apply';
            app.ApplyTemperatureButton.Position = [47 124 100 22];
            % Create SharpnessTab
            app.SharpnessTab = uitab(app.TabGroup3);
            app.SharpnessTab.Title = 'Sharpness';
            % Create SharpnessSlider
            app.SharpnessSlider = uislider(app.SharpnessTab);
            app.SharpnessSlider.Limits = [0 5];
            app.SharpnessSlider.MajorTicks = 0;
            app.SharpnessSlider.MajorTickLabels = {};
            app.SharpnessSlider.ValueChangedFcn = createCallbackFcn(app,
@SharpnessSliderValueChanged, true);
            app.SharpnessSlider.MinorTicks = [];
            app.SharpnessSlider.Position = [14 173 166 3];
            % Create ApplySharpnessButton
            app.ApplySharpnessButton = uibutton(app.SharpnessTab, 'state');
            app.ApplySharpnessButton.ValueChangedFcn = createCallbackFcn(app,
@ApplySharpnessButtonValueChanged, true);
            app.ApplySharpnessButton.Text = 'Apply';
            app.ApplySharpnessButton.Position = [47 124 100 22];
            % Create HistogramTab
            app.HistogramTab = uitab(app.TabGroup3);
            app.HistogramTab.Title = 'Histogram';
            % Create HistogramAxes
            app.HistogramAxes = uiaxes(app.HistogramTab);
            app.HistogramAxes.Box = 'on';
            app.HistogramAxes.XGrid = 'on';
            app.HistogramAxes.YGrid = 'on';
            app.HistogramAxes.Position = [5 72 188 119];
            % Create AutoAdjustButton
```

```
            app.AutoAdjustButton = uibutton(app.ColorAdjustTab, 'push');
            app.AutoAdjustButton.ButtonPushedFcn = createCallbackFcn(app,
@AutoAdjustButtonPushed, true);
            app.AutoAdjustButton.Position = [104 232 100 22];
            app.AutoAdjustButton.Text = 'Auto Adjust';
            % Create OverlayTab
            app.OverlayTab = uitab(app.TabGroup);
            app.OverlayTab.Title = 'Overlay';
            % Create FiltersListBoxLabel
            app.FiltersListBoxLabel = uilabel(app.OverlayTab);
            app.FiltersListBoxLabel.BackgroundColor = [0.502 0.502 0.502];
            app.FiltersListBoxLabel.HorizontalAlignment = 'center';
            app.FiltersListBoxLabel.FontColor = [1 1 1];
            app.FiltersListBoxLabel.Position = [137 208 39 22];
            app.FiltersListBoxLabel.Text = 'Filters';
            % Create FiltersListBox
            app.FiltersListBox = uilistbox(app.OverlayTab);
            app.FiltersListBox.Items = {'None', 'Black & White', 'Sepia', 'Negative',
'Winter'};
            app.FiltersListBox.ValueChangedFcn = createCallbackFcn(app,
@FiltersListBoxValueChanged, true);
            app.FiltersListBox.FontColor = [1 1 1];
            app.FiltersListBox.BackgroundColor = [0.502 0.502 0.502];
            app.FiltersListBox.Position = [93 97 128 98];
            app.FiltersListBox.Value = 'None';
            % Create ApplyFilterButton
            app.ApplyFilterButton = uibutton(app.OverlayTab, 'push');
            app.ApplyFilterButton.ButtonPushedFcn = createCallbackFcn(app,
@ApplyFilterButtonPushed, true);
            app.ApplyFilterButton.Position = [107 45 100 22];
            app.ApplyFilterButton.Text = 'Apply';
            % Create HistoryTab
            app.HistoryTab = uitab(app.TabGroup);
            app.HistoryTab.Title = 'History';
            % Create ListBox
            app.ListBox = uilistbox(app.HistoryTab);
            app.ListBox.Items = {'Original Image', 'Cropped Image', 'Rotated Image',
'Brightness', 'Contrast', 'Saturation', 'Temperature', 'Sharpness', 'Filter Applied Image',
'Auto Adjusted Image'};
            app.ListBox.ValueChangedFcn = createCallbackFcn(app, @ListBoxValueChanged, true);
            app.ListBox.Position = [89 53 142 189];
            app.ListBox.Value = 'Original Image';
            % Create RollBackButton
            app.RollBackButton = uibutton(app.HistoryTab, 'push');
            app.RollBackButton.ButtonPushedFcn = createCallbackFcn(app,
@RollBackButtonPushed, true);
            app.RollBackButton.Position = [107 19 100 22];
            app.RollBackButton.Text = 'Roll Back';
        end
    end
    methods (Access = public)
        % Construct app
        function app = app1
            % Create and configure components
            createComponents(app)
```

```
            % Register the app with App Designer
            registerApp(app, app.PhotoEditor)
            if nargout == 0
                clear app
            end
        end
        % Code that executes before app deletion
        function delete(app)
            % Delete UIFigure when app is deleted
            delete(app.PhotoEditor)
        end
    end
end
```