****

**Mục Lục**

[**1.** **Đọc ảnh** 3](#_Toc123806214)

[1.1 Thông tin ảnh. 3](#_Toc123806215)

[**2.** **Chuyển đổi ảnh** 4](#_Toc123806216)

[**3.** **Thao tác với ảnh màu RGB** 5](#_Toc123806217)

[**4.** **Xử lý điểm:** 7](#_Toc123806218)

[- **Ảnh âm bản** 7](#_Toc123806219)

[- **Phân ngưỡng đơn trên ảnh** 8](#_Toc123806220)

[- **Phân ngưỡng kép trên ảnh** 10](#_Toc123806221)

[**5.** **Xử lý tổ chức đồ:** 13](#_Toc123806222)

[- **Hiển thị tổ chức đồ của ảnh** 13](#_Toc123806223)

[- **Trượt Histogram:** 13](#_Toc123806224)

[***6.*** **Các phép toán số học trên ảnh *(có thể yêu cầu tự viết hàm hoặc dùng hàm dựng sẵn)*** 23](#_Toc123806225)

[**7.** **Lọc làm mịn ảnh: mean, gaussian, median, order filter *(sử dụng các hàm lọc ảnh dựng sẵn)*** 30](#_Toc123806226)

[**8.** **Phát hiện biên + lọc làm nét ảnh: Robert, Prewitt, Sobel, Laplacian *(chỉ sử dụng các hàm lọc ảnh dựng sẵn)*** 34](#_Toc123806227)

[**9.** **Xử lý hình thái** 37](#_Toc123806228)

[**10.** **Sử dụng các components trong component library của MATLAB App Designer để tuỳ biến các thông số, bộ lọc; thiết kế giao diện đảm bảo tính tương tác tốt với người dùng.** 39](#_Toc123806229)

**NỘI DUNG THI HỌC PHẦN XỬ LÝ ẢNH**

**Xây dựng App trong App Designer của Matlab**

1. **Đọc ảnh**

[filename,path]=uigetfile({'\*.png';'\*.jpg';'\*.tif'},"chọn ảnh");

if isequal(filename,0)||isequal(path,0)

uialert(app.UIFigure,"Chưa đọc ảnh","Lỗi đọc ảnh");

cla(app.UIAnhGoc);

else

file=strcat(path,filename);

img=imread(file);

imshow(img,'Parent',app.UIAnhGoc);

end

## Thông tin ảnh.

info=imfinfo(file);

arr=[];

if length(info)>1

app.ThngtinnhTextAreaLabel.Text=strcat(strcat("Tập tin chứa: ",num2str(length(info)))," ảnh");

arr=[arr;"THÔNG TIN ẢNH ĐẦU TIÊN"];

info=info(1);

else

app.ThngtinnhTextAreaLabel.Text="";

end

app.colortype=info.ColorType;

fields=fieldnames(info);

for i=1:length(fields)

if isnumeric(info.(char(fileds(i))))

value=num2str(info.(char(fields(i))));

else

value=info.(char(fields(i)));

end

str=strcat(fields(i),":",value);

arr=[arr;str];

arr=[arr;'----------------'];

end

app.txtAThongTinAnh.Value=arr;

end

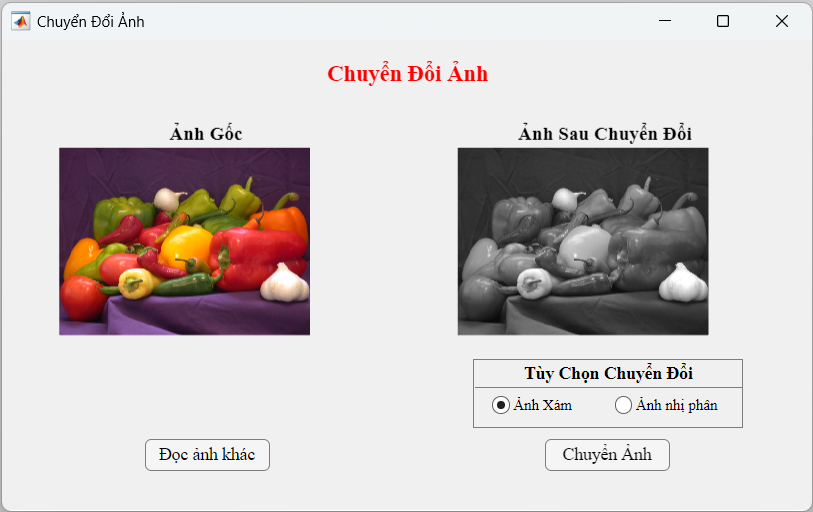
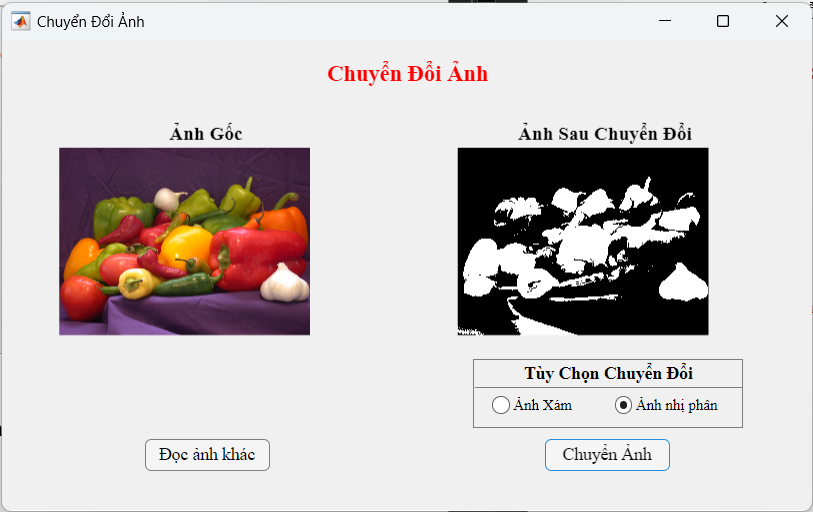
imshow(gray,'Parent',app.UIAnhXam);

else

imshow(app.img,'Parent',app.UIAnhXam);

end

1. **Chuyển đổi ảnh**

**+ Chuyển gray và binary.**

Bỏ check thuộc tích Enable của Radio Button Group nhằm ngăn người dùng thao tác trên dữ liệu rỗng.

**Xây dựng giao diện:**

• **Button**: app.btn\_DocAnh

• **Panel** app.pn\_Main:

o **Axes**: app.axes\_AnhGoc, app.axes\_AnhChuyenDoi

o **Button**: app.btn\_Chuyen

o **Radio Button Group**: rg\_TuyChon, gồm các Radio Button: app.rb\_AnhXam,

app.rb\_NhiPhan

****

properties (Access = private)

img

map

colortype

img\_cd

end

****

[filename, path] = uigetfile({'\*.png'; '\*.jpg';'\*.tif'}, 'Chọn ảnh');

if ~isequal(filename, 0) && ~isequal(path, 0)

file = strcat(path, filename);

[app.img, app.map] = imread(file);

app.colortype = imfinfo(file).ColorType;

imshow(app.img, app.map, 'Parent', app.UIAxes\_AnhGoc);

app.Btn\_DocAnh.Text = "Đọc ảnh khác";

app.RadioGroup.Enable = "on";

end

****

if app.colortype == "truecolor"

app.img\_cd = rgb2gray(app.img);

elseif app.colortype == "indexed"

app.img\_cd = ind2gray(app.img, app.map);

else

app.img\_cd = app.img;

end

if app.RadioGroup.SelectedObject == app.Btn\_AnhNhiPhan

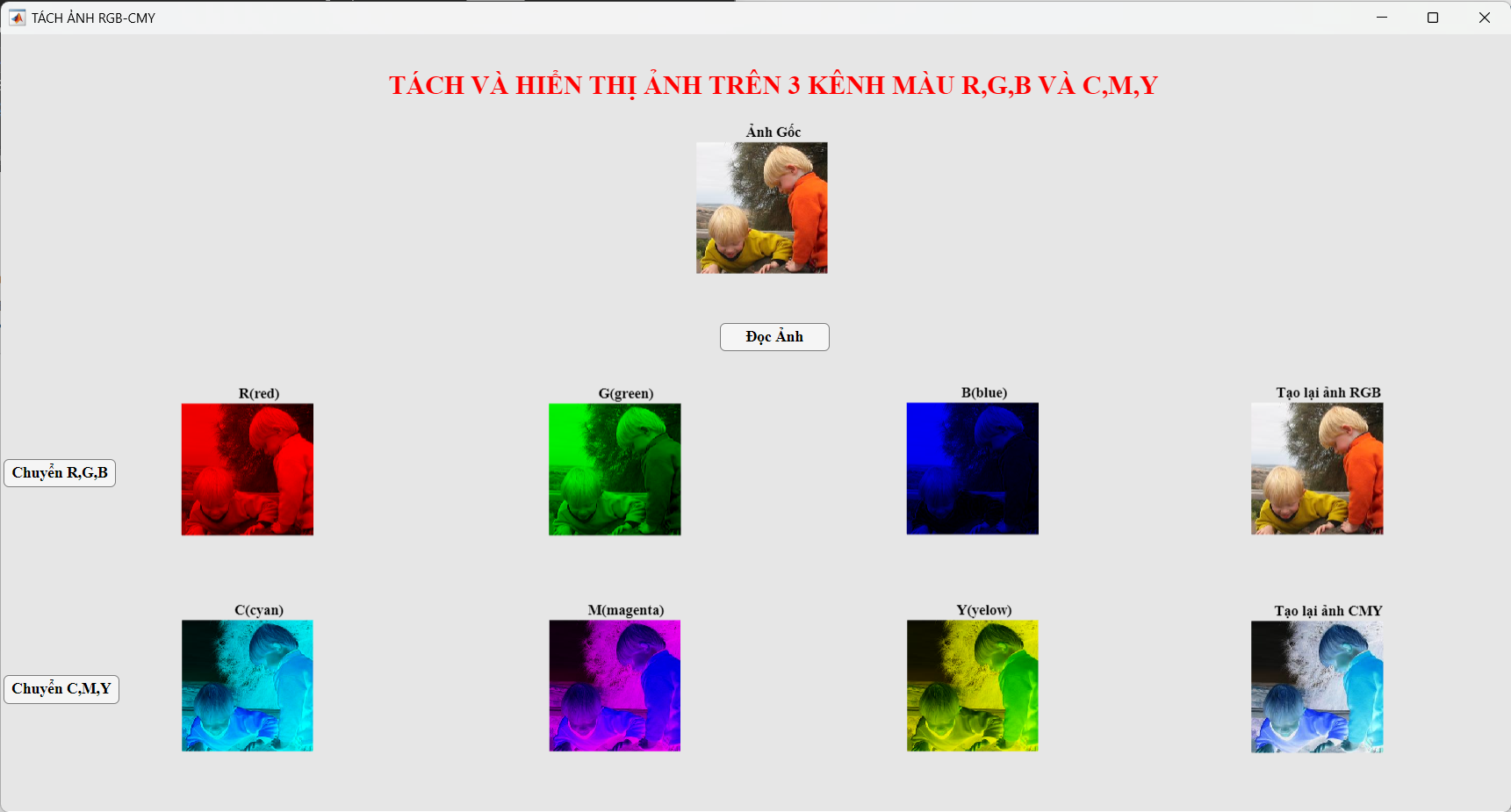
app.img\_cd = imbinarize(app.img\_cd);

end

imshow(app.img\_cd,'Parent',app.UIAxes\_KetQua);

1. **Thao tác với ảnh màu RGB**

* **Tách và hiển thị ảnh trên 3 kênh màu R, G, B hoặc trên 3 kênh màu bù C, M, Y**
* **Tạo lại ảnh RGB**
* **Tạo ảnh CMY**



* Khai báo:

properties (Access = private)

img

map

colortype

gray

end

* Btn\_DocAnh:

[filename,path]=uigetfile({'\*.png';'\*.jpg';'\*.tif'},"chọn ảnh");

if isequal(filename,0)||isequal(path,0)

uialert(app.UIFigure,"Chưa đọc ảnh","Lỗi đọc ảnh");

cla(app.Axes\_AnhGoc);

else

file=strcat(path,filename);

app.img=imread(file);

info=imfinfo(file);

app.colortype=info.ColorType;

if app.colortype=="truecolor"

imshow(app.img,'Parent',app.Axes\_AnhGoc);

set(app.Btn\_RGB,'visible','on');

set(app.Btn\_CMY,'visible','on');

else

uialert(app.UIFigure,"Ảnh không thuộc định dạng RGB","Lỗi");

set(app.Btn\_RGB,'visible','off');

set(app.Btn\_CMY,'visoble','off');

end

end

* Btn\_RGB:

RGB= app.img;

red\_img=RGB;

red\_img(:,:,2)=0;

red\_img(:,:,3)=0;

green\_img=RGB;

green\_img(:,:,1)=0;

green\_img(:,:,3)=0;

blue\_img=RGB;

blue\_img(:,:,1)=0;

blue\_img(:,:,2)=0;

imshow(RGB,'Parent',app.Axes\_TaoLaiRGB);

imshow(red\_img,'Parent',app.Axes\_Red);

imshow(green\_img,'Parent',app.Axes\_Green);

imshow(blue\_img,'Parent',app.Axes\_Blue);

* Btn\_CMY:

CMY=app.img;

CMY(:,:,1)=255-CMY(:,:,1);

CMY(:,:,2)=255-CMY(:,:,2);

CMY(:,:,3)=255-CMY(:,:,3);

cyan=CMY;

magenta=CMY;

yellow=CMY;

cyan(:,:,1)=0;

magenta(:,:,2)=0;

yellow(:,:,3)=0;

imshow(CMY,'Parent',app.Axes\_TaoLaiCMY);

imshow(cyan,'Parent',app.Axes\_Cyan);

imshow(magenta,'Parent',app.Axes\_Magenta);

imshow(yellow,'Parent',app.Axes\_Yelow);

1. **Xử lý điểm:**

* **Ảnh âm bản**

****



properties (Access = private)

img

map

end



[filename, path] = uigetfile({'\*.png'; '\*.jpg';'\*.tif'}, 'Chọn ảnh');

if ~isequal(filename, 0) && ~isequal(path, 0)

file = strcat(path, filename);

[app.img, app.map] = imread(file);

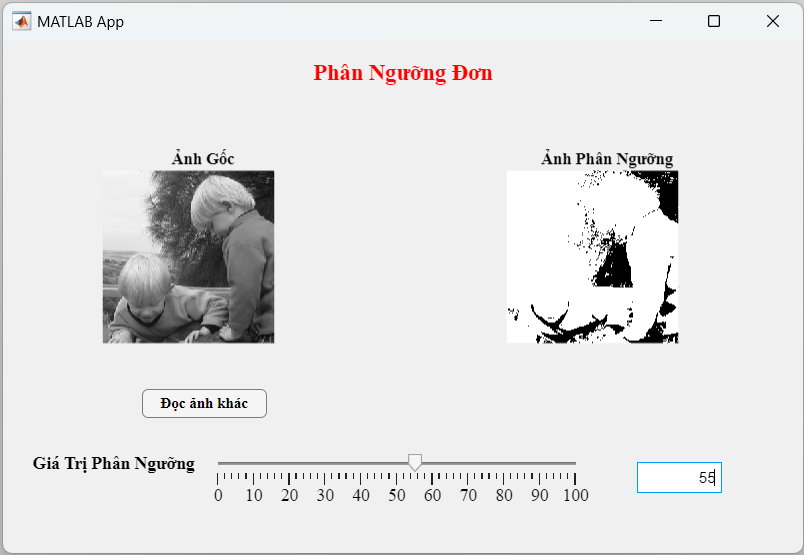
imshow(app.img, app.map, 'Parent', app.UIAxes\_AnhGoc);

imshow(imcomplement(app.img), app.map, 'Parent',app.UIAxes\_KetQua);

app.Btn\_DocAnh.Text = "Đọc ảnh khác";

end

* **Phân ngưỡng đơn trên ảnh**

****

**Xây dựng giao diện:**

• **Axes:** app.axs\_AnhGoc, app.axs\_AnhPhanNguong

• **Button**: app.btn\_DocAnh

• **Slider:** app.sld\_GiaTri

• **Edit Field:** app.ef\_GiaTri



properties (Access = private)

img

img\_gray

map

info

colortype

end



[filename, path] = uigetfile({'\*.png'; '\*.jpg'; '\*.tif'}, 'Chọn ảnh');

if ~isequal(filename, 0) && ~isequal(path, 0)

file = strcat(path, filename);

app.info = imfinfo(file);

if length(app.info) > 1

app.info = app.info(1);

end

Lmax = 2^app.info.BitDepth - 1;

if Lmax == 1

uialert(app.UIFigure, 'Ảnh vừa chọn là ảnh nhị phân', 'Chú ý');

return;

end

[app.img, app.map] = imread(file);

app.colortype = app.info.ColorType;

if app.colortype == "truecolor"

app.img\_gray = rgb2gray(app.img);

elseif app.colortype == "indexed"

app.img\_gray = ind2gray(app.img, app.map);

else

app.img\_gray = app.img;

end

imshow(app.img\_gray, app.map, 'Parent', app.UIAxes\_AnhGoc);

imshow(app.img\_gray > 128, app.map, 'Parent',app.UIAxes\_KetQua);

app.Btn\_DocAnh.Text = "Đọc ảnh khác";

app.Slider\_PhanNguong.Enable = "on";

app.Ed\_GiaTri.Enable = "on";

app.Ed\_GiaTri.Value = 100;

app.Slider\_PhanNguong.Value = 100;

end



changingValue = event.Value;

app.Ed\_GiaTri.Value = round(changingValue, 0);

imshow(app.img\_gray > app.Ed\_GiaTri.Value, 'Parent',app.UIAxes\_KetQua);



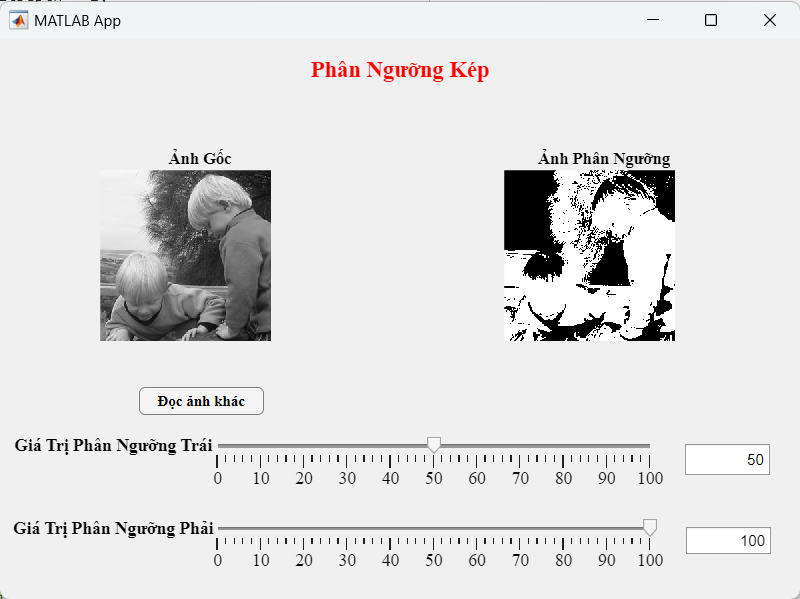
value = app.Ed\_GiaTri.Value;

app.Ed\_GiaTri.Value = round(value, 0);

app.Slider\_PhanNguong.Value = round(value, 0);

imshow(app.img\_gray > app.Ed\_GiaTri.Value, 'Parent', app.UIAxes\_KetQua);

* **Phân ngưỡng kép trên ảnh**



**Xây dựng giao diện:**

• **Axes:** app.axes\_AnhGoc, app.axes\_AnhPhanNguong

• **Button**: app.btn\_DocAnh

• **Slider**: app.sld\_NguongTrai, app.sld\_NguongPhai

• **Edit Field:** app.ef\_NguongTrai, app.ef\_NguongPhai



properties (Access = private)

img

map

info

colortype

img\_gray

end

****

[filename, path] = uigetfile({'\*.png'; '\*.jpg'; '\*.tif'}, 'Chọn ảnh');

if isequal(filename, 0) || isequal(path, 0)

return;

end

file = strcat(path, filename);

app.info = imfinfo(file);

if length(app.info) > 1

app.info = app.info(1);

end

Lmax = 2^app.info.BitDepth - 1;

if Lmax == 1

uialert(app.UIFigure, 'Ảnh vừa chọn là ảnh nhị phân', 'Chú ý');

return;

end

[app.img, app.map] = imread(file);

app.colortype = app.info.ColorType;

if app.colortype == "truecolor"

app.img\_gray = rgb2gray(app.img);

elseif app.colortype == "indexed"

app.img\_gray = ind2gray(app.img, app.map);

else

app.img\_gray = app.img;

end

imshow(app.img\_gray, app.map, 'Parent', app.UIAxes\_AnhGoc);

imshow(app.img\_gray > 64 & app.img\_gray < 192, app.map, 'Parent',app.UIAxes\_KetQua);

app.Btn\_DocAnh.Text = "Đọc ảnh khác";

app.Slider\_PhanNguongTrai.Enable = "on";

app.Slider\_PhanNguongPhai.Enable = "on";

app.Ed\_GiaTriTrai.Enable = "on";

app.Ed\_GiaTriPhai.Enable = "on";

app.Slider\_PhanNguongTrai.Value = 50;

app.Slider\_PhanNguongPhai.Value = 100;

app.Ed\_GiaTriTrai.Value = 50;

app.Ed\_GiaTriPhai.Value = 100;

****

changingValue = event.Value;

app.Ed\_GiaTriTrai.Value = round(changingValue, 0);

if app.Ed\_GiaTriTrai.Value > app.Ed\_GiaTriPhai.Value

app.Ed\_GiaTriPhai.Value = app.Ed\_GiaTriTrai.Value;

app.Slider\_PhanNguongPhai.Value = app.Ed\_GiaTriTrai.Value;

end

imshow(app.img\_gray > app.Ed\_GiaTriTrai.Value & app.img\_gray < app.Ed\_GiaTriPhai.Value, 'Parent', app.UIAxes\_KetQua);

****

changingValue = event.Value;

app.Ed\_GiaTriPhai.Value = round(changingValue, 0);

if app.Ed\_GiaTriPhai.Value < app.Ed\_GiaTriTrai.Value

app.Ed\_GiaTriTrai.Value = app.Ed\_GiaTriPhai.Value;

app.Slider\_PhanNguongTrai.Value = app.Ed\_GiaTriPhai.Value;

end

imshow(app.img\_gray > app.Ed\_GiaTriTrai.Value & app.img\_gray < app.Ed\_GiaTriPhai.Value, 'Parent', app.UIAxes\_KetQua);

****

value = app.Ed\_GiaTriTrai.Value;

app.Ed\_GiaTriTrai.Value = round(value, 0);

app.Slider\_PhanNguongTrai.Value = round(value, 0);

if app.Ed\_GiaTriTrai.Value > app.Ed\_GiaTriPhai.Value

app.Ed\_GiaTriPhai.Value = app.Ed\_GiaTriTrai.Value;

app.Slider\_PhanNguongPhai.Value = app.Ed\_GiaTriTrai.Value;

end

imshow(app.img\_gray > app.Ed\_GiaTriTrai.Value & app.img\_gray < app.Ed\_GiaTriPhai.Value, 'Parent', app.UIAxes\_KetQua);

****

value = app.Ed\_GiaTriPhai.Value;

app.Ed\_GiaTriPhai.Value = round(value, 0);

app.Slider\_PhanNguongPhai.Value = round(value, 0);

if app.Ed\_GiaTriPhai.Value < app.Ed\_GiaTriTrai.Value

app.Ed\_GiaTriTrai.Value = app.Ed\_GiaTriPhai.Value;

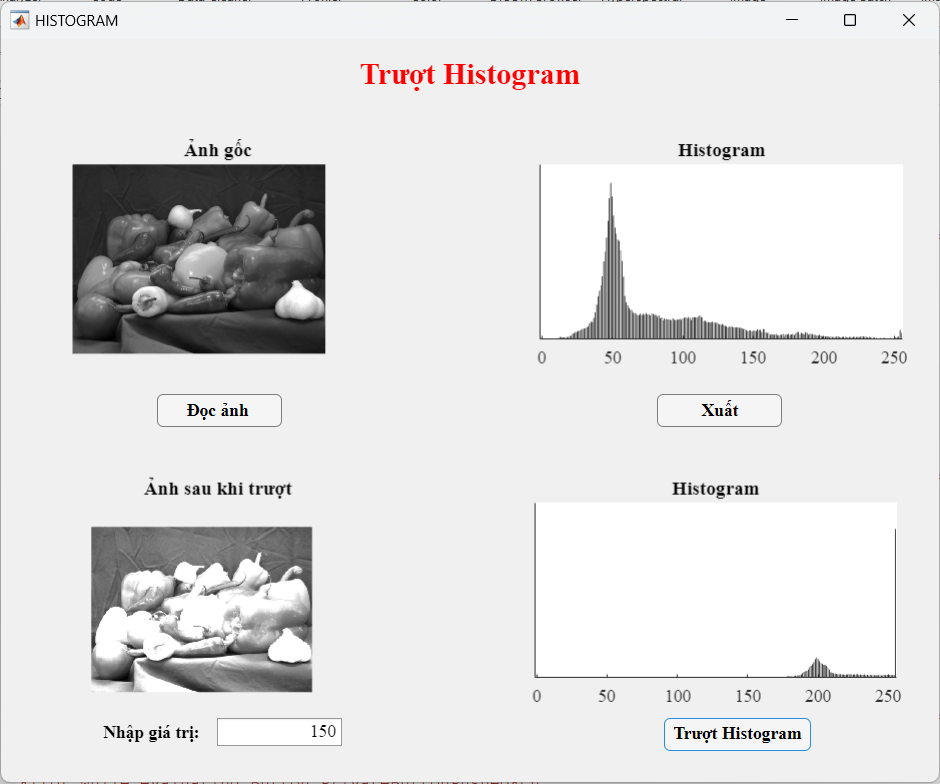
app.Slider\_PhanNguongTrai.Value = app.Ed\_GiaTriPhai.Value;

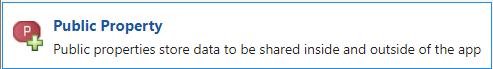
end

imshow(app.img\_gray > app.Ed\_GiaTriTrai.Value & app.img\_gray < app.Ed\_GiaTriPhai.Value, 'Parent', app.UIAxes\_KetQua);

1. **Xử lý tổ chức đồ:**

* **Hiển thị tổ chức đồ của ảnh**
* **Trượt Histogram:**



****

properties (Access = private)

Property

img

map

colortype

end

properties (Access = public)

bitdepth

end

methods (Access = private)

function oimg = TruotHistogram(app)

[y,x]=size(app.img);

oimg=zeros(y,x);

c=app.EditField.Value;

Lmax=2^app.bitdepth-1;

for i=1:y

for j=1:x

oimg(i,j)=app.img(i,j)+c;

if(oimg(i,j)>Lmax)

oimg(i,j)=Lmax;

end

if oimg(i,j)<0

oimg(i,j)=0;

end

end

end

end

end

****

[filename, path]=uigetfile({'\*.jpg';'\*.png';'\*.tif'},"Chọn tập tin ảnh");

if isequal(filename,0)||isequal(path,0)

uialert(app.UIFigure,"Chưa đọc tập tin ảnh","Lỗi đọc ảnh");

cla(app.UIXes\_AnhGoc);

else

file=strcat(path, filename);

info=imfinfo(file);

if length(info)>1

info=info(1);

end

app.colortype=info.ColorType;

app.bitdepth=info.BitDepth;

[app.img,app.map]=imread(file);

if app.colortype=="truecolor"

app.img=rgb2gray(app.img);

elseif app.colortype=="indexed"

app.img=ind2gray(app.img);

end

imshow(app.img,[],'Parent',app.UIXes\_AnhGoc);

end

****

[pixels,graylevels]=imhist(app.img);

bar(app.UIXes\_HienThi,graylevels,pixels,0.5,"stacked","black");



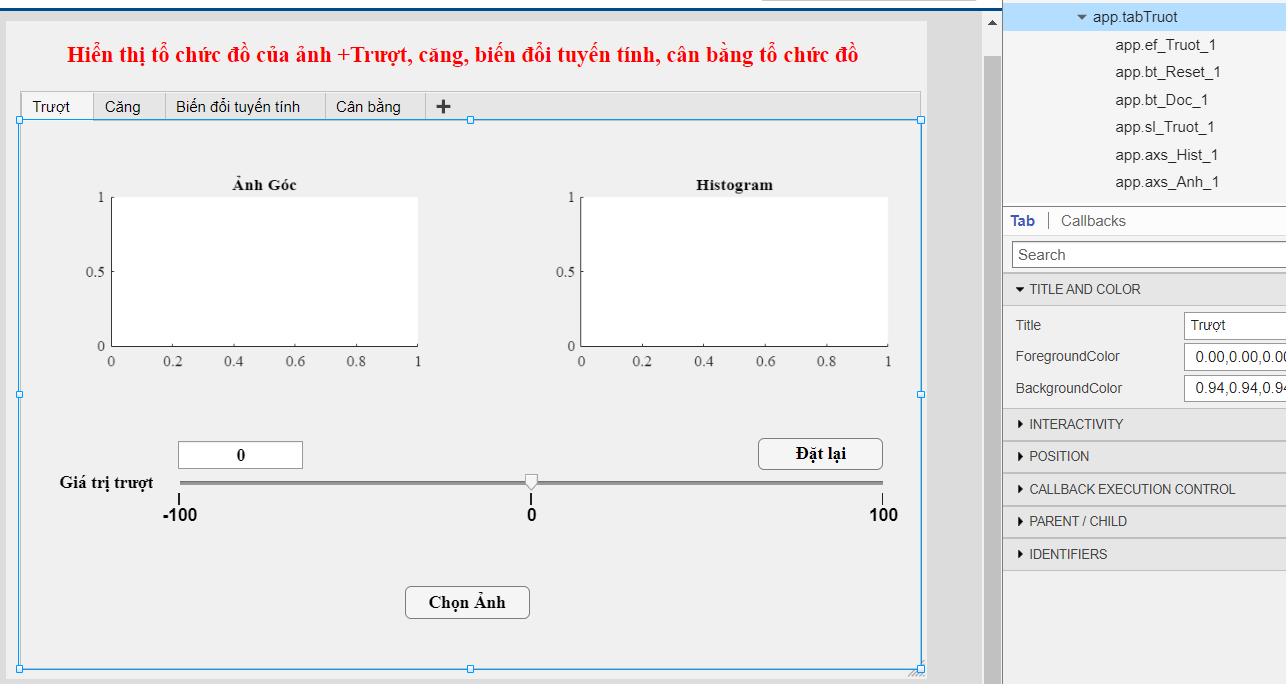
oimg=app.TruotHistogram;

imshow(oimg,[],'Parent',app.anhsaukhitruot);

[onumberofpixel, ograyscale]=imhist(uint8(oimg));

bar(app.histogramtruot, ograyscale,onumberofpixel,0.5,"stacked","black");

* **Trượt, căng, biến đổi tuyến tính, cân bằng tổ chức đồ *(có thể yêu cầu tự viết hàm hoặc dùng hàm dựng sẵn)***



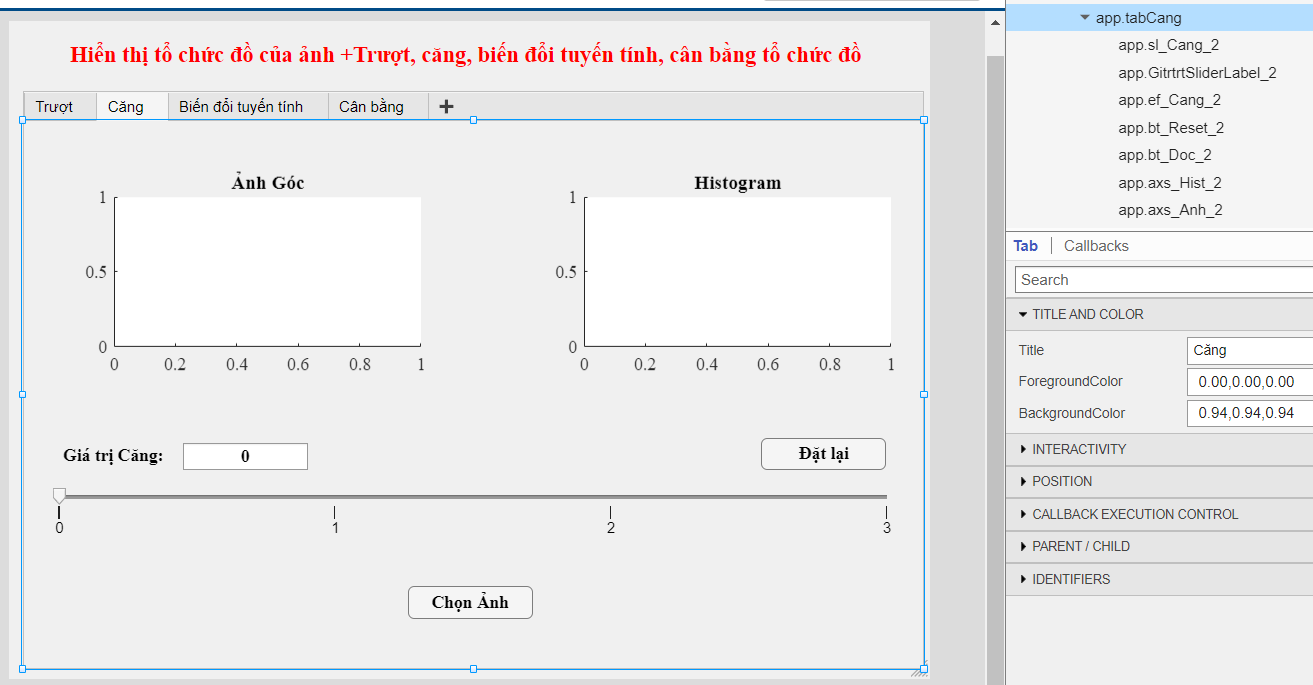
**• app.tabTruot (Trượt)**

o **Axes**: app.axs\_Anh\_1, app.axs\_Hist\_1

o **Button**: app.bt\_Doc\_1, app.bt\_Reset\_1

o **Slider**: app.sl\_Truot\_1

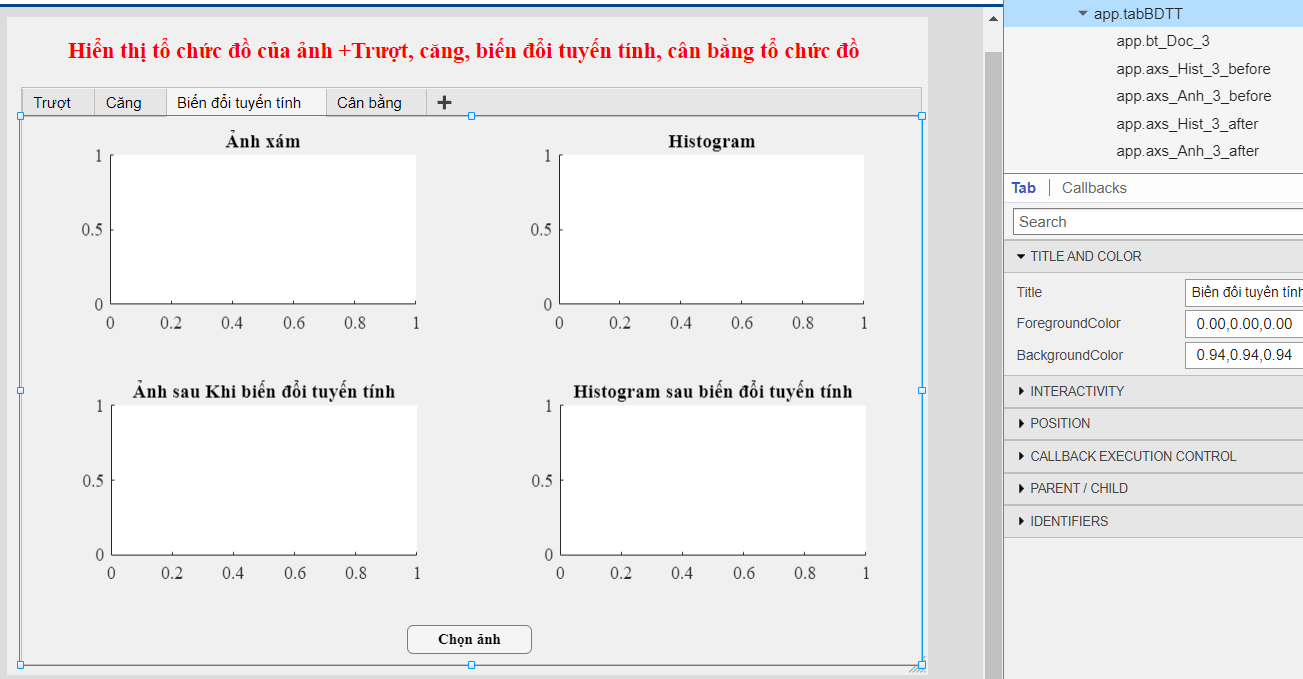
o **Edit Field**: app.ef\_Truot\_1

****

**• app.tabDBTT (Biến đổi tuyến tính)**

o **Axes**: app.axs\_Anh\_3\_before, app.axs\_Anh\_3\_after, app.axs\_Hist\_3\_before,app.axs\_Hist\_3\_after

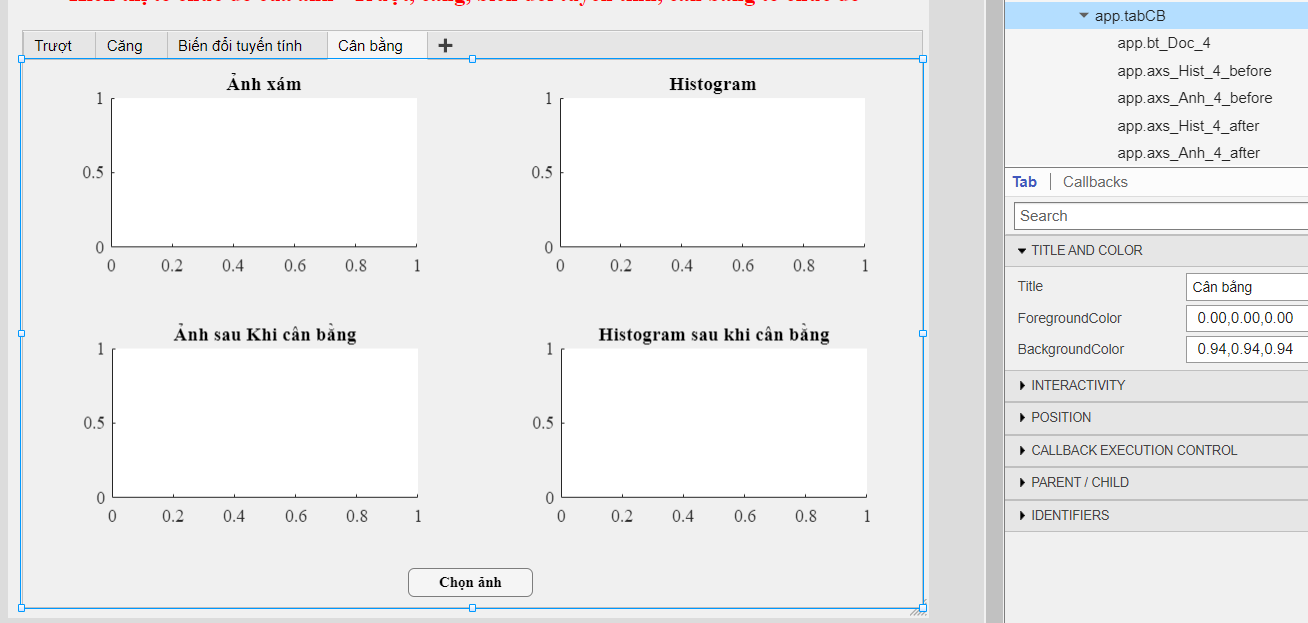
o **Button**: app.bt\_Doc\_3



**• app.tabDBTT (Biến đổi tuyến tính)**

o **Axes:** app.axs\_Anh\_3\_before, app.axs\_Anh\_3\_after, app.axs\_Hist\_3\_before, app.axs\_Hist\_3\_after

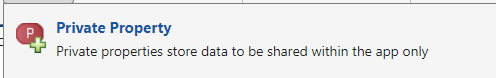
o **Button:** app.bt\_Doc\_3



**• app.tabCB (Cân bằng)**

o **Axes**: app.axs\_Anh\_4\_before, app.axs\_Anh\_4\_after, app.axs\_Hist\_4\_before, app.axs\_Hist\_4\_after.

o **Button**: app.bt\_Doc\_4.



properties (Access = private)

% \_1: thuộc tab Trượt; \_2: thuộc tab Căng; \_3: thuộc tab Biến đổi tuyến tính; \_4: thuộc tab Cân bằng

img\_1

info\_1

gray\_1

map\_1

colortype\_1

Lmax\_1

img\_2

info\_2

gray\_2

map\_2

colortype\_2

Lmax\_2

img\_3

info\_3

gray\_3

map\_3

colortype\_3

Lmax\_3

img\_4

info\_4

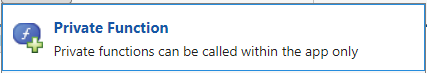
gray\_4

map\_4

colortype\_4

Lmax\_4

end



methods (Access = private)

function oimg = truotToChucDo(app)

[y, x] = size(app.gray\_1);

oimg = zeros(y, x);

c = app.ef\_Truot\_1.Value;

for i = 1 : y

for j = 1 : x

oimg(i, j) = app.gray\_1(i, j) + c;

if(oimg(i, j) > app.Lmax\_1)

oimg(i, j) = app.Lmax\_1;

elseif (oimg(i, j) < 0)

oimg(i, j) = 0;

end

end

end

end

function oimg = cangToChucDo(app)

[y, x] = size(app.gray\_2);

oimg = zeros(y, x);

c = app.ef\_Cang\_2.Value;

for i = 1 : y

for j = 1 : x

oimg(i, j) = app.gray\_2(i, j) \* c;

if(oimg(i, j) > app.Lmax\_2)

oimg(i, j) = app.Lmax\_2;

elseif (oimg(i, j) < 0)

oimg(i, j) = 0;

end

end

end

end

end



f = app.axs\_Anh\_1;

[filename, path] = uigetfile({'\*.png'; '\*.jpg'; '\*.tif'}, 'Chọn ảnh');

if isequal(filename, 0) || isequal(path, 0)

uialert(f, 'Chưa chọn ảnh', 'Lỗi đọc ảnh');

cla(app.axs\_Anh\_1);

else

% Enable các thành phần cần thiết

app.sl\_Truot\_1.Enable = true;

app.ef\_Truot\_1.Enable = true;

app.bt\_Reset\_1.Visible = true;

app.bt\_Doc\_1.Text = 'Chọn ảnh khác';

% Đọc ảnh đã chọn và xuất ảnh xám

file = strcat(path, filename);

[app.img\_1, app.map\_1] = imread(file);

app.info\_1 = imfinfo(file);

app.colortype\_1 = app.info\_1.ColorType;

if app.colortype\_1 == "truecolor"

app.gray\_1 = rgb2gray(app.img\_1);

elseif app.colortype\_1 == "indexed"

app.gray\_1 = ind2gray(app.img\_1, app.map\_1);

else

app.gray\_1 = app.img\_1;

end

imshow(app.gray\_1, app.map\_1, 'Parent', app.axs\_Anh\_1);

if length(app.info\_1) > 1

app.info\_1 = app.info\_1(1);

end

app.Lmax\_1 = 2^app.info\_1.BitDepth - 1;

if app.Lmax\_1 == 1

uialert(f, 'Ảnh vừa được chọn là ảnh nhị phân!', 'Lưu ý');

end

% Xuất hist gốc

app.axs\_Hist\_1.reset;

[numberofpixel, grayscale] = imhist(app.gray\_1);

bar(app.axs\_Hist\_1, grayscale, numberofpixel, 0.5, "stacked", "black");

app.ef\_Truot\_1.Value = 0;

app.sl\_Truot\_1.Value = 0;

end



changingValue = event.Value;

% Đồng bộ dữ liệu với Edit field

app.ef\_Truot\_1.Value = round(changingValue, 0);

% Xử lý

oimg = app.truotToChucDo;

% Xuất thành quả: ảnh output và hist của nó

imshow(oimg , [], 'Parent', app.axs\_Anh\_1);[onumberofpixel, ograyscale] = imhist(uint8(oimg));

bar(app.axs\_Hist\_1, ograyscale, onumberofpixel, 0.5, "stacked", "blue");



value = app.ef\_Truot\_1.Value;

% Đồng bộ dữ liệu với Slider

app.sl\_Truot\_1.Value = value;

% Xử lý

oimg = app.truotToChucDo;

% Xuất thành quả: ảnh output và hist của nó

imshow(oimg , [], 'Parent', app.axs\_Anh\_1);

[onumberofpixel, ograyscale] = imhist(uint8(oimg));

bar(app.axs\_Hist\_1, ograyscale, onumberofpixel, 0.5, "stacked", "blue");



imshow(app.gray\_1, app.map\_1, 'Parent', app.axs\_Anh\_1);

% Xuất hist gốc

app.axs\_Hist\_1.reset;

[numberofpixel, grayscale] = imhist(app.gray\_1);

bar(app.axs\_Hist\_1, grayscale, numberofpixel, 0.5, "stacked", "black");

app.ef\_Truot\_1.Value = 0;

app.sl\_Truot\_1.Value = 0;



f = app.axs\_Anh\_2;

[filename, path] = uigetfile({'\*.png'; '\*.jpg'; '\*.tif'}, 'Chọn ảnh');

if isequal(filename, 0) || isequal(path, 0)

uialert(f, 'Chưa chọn ảnh', 'Lỗi đọc ảnh');

cla(app.axs\_Anh\_2);

else

% Enable các thành phần cần thiết

app.sl\_Cang\_2.Enable = true;

app.ef\_Cang\_2.Enable = true;

app.bt\_Reset\_2.Visible = true;

app.bt\_Doc\_2.Text = 'Chọn ảnh khác';

% Đọc ảnh đã chọn và xuất ảnh xám

file = strcat(path, filename);

[app.img\_2, app.map\_2] = imread(file);

app.info\_2 = imfinfo(file);

app.colortype\_2 = app.info\_2.ColorType;

if app.colortype\_2 == "truecolor"

app.gray\_2 = rgb2gray(app.img\_2);

elseif app.colortype\_2 == "indexed"

app.gray\_2 = ind2gray(app.img\_2, app.map\_2);

else

app.gray\_2 = app.img\_2;

end

imshow(app.gray\_2, app.map\_2, 'Parent', app.axs\_Anh\_2);

if length(app.info\_2) > 1

app.info\_2 = app.info\_2(1);

end

app.Lmax\_2 = 2^app.info\_2.BitDepth - 1;

if app.Lmax\_2 == 1

uialert(f, 'Ảnh vừa được chọn là ảnh nhị phân!', 'Lưu ý');

end

% Xuất hist gốc

app.axs\_Hist\_2.reset;

[numberofpixel, grayscale] = imhist(app.gray\_2);

bar(app.axs\_Hist\_2, grayscale, numberofpixel, 0.5, "stacked", "black");

app.ef\_Cang\_2.Value = 1;

app.sl\_Cang\_2.Value = 1;

end



changingValue = event.Value;

% Đồng bộ dữ liệu với Edit field

app.ef\_Cang\_2.Value = changingValue;

% Xử lý

oimg = app.cangToChucDo;

% Xuất thành quả: ảnh output và hist của nó

imshow(oimg , [], 'Parent', app.axs\_Anh\_2);

[onumberofpixel, ograyscale] = imhist(uint8(oimg));

bar(app.axs\_Hist\_2, ograyscale, onumberofpixel, 0.5, "stacked", "blue");



value = app.ef\_Cang\_2.Value;

% Đồng bộ dữ liệu với Slider

app.sl\_Cang\_2.Value = value;

% Xử lý

oimg = app.cangToChucDo;

% Xuất thành quả: ảnh output và hist của nó

imshow(oimg , [], 'Parent', app.axs\_Anh\_2);

[onumberofpixel, ograyscale] = imhist(uint8(oimg));

bar(app.axs\_Hist\_2, ograyscale, onumberofpixel, 0.5, "stacked", "blue");



imshow(app.gray\_2, app.map\_2, 'Parent', app.axs\_Anh\_2);

% Xuất hist gốc

app.axs\_Hist\_2.reset;

[numberofpixel, grayscale] = imhist(app.gray\_2);

bar(app.axs\_Hist\_2, grayscale, numberofpixel, 0.5, "stacked", "black");

app.ef\_Cang\_2.Value = 1;

app.sl\_Cang\_2.Value = 1;



f = app.axs\_Anh\_3\_after;

[filename, path] = uigetfile({'\*.png'; '\*.jpg'; '\*.tif'}, 'Chọn ảnh');

if isequal(filename, 0) || isequal(path, 0)

uialert(f, 'Chưa chọn ảnh', 'Lỗi đọc ảnh');

cla(app.axs\_Anh\_3\_before);

else

% Enable các thành phần cần thiết

app.bt\_Doc\_3.Text = 'Chọn ảnh khác';

% Đọc ảnh đã chọn và xuất ảnh xám

file = strcat(path, filename);

[app.img\_3, app.map\_3] = imread(file);

app.info\_3 = imfinfo(file);

app.colortype\_3 = app.info\_3.ColorType;

if app.colortype\_3 == "truecolor"

app.gray\_3 = rgb2gray(app.img\_3);

elseif app.colortype\_3 == "indexed"

app.gray\_3 = ind2gray(app.img\_3, app.map\_3);

else

app.gray\_3 = app.img\_3;

end

imshow(app.gray\_3, app.map\_3, 'Parent', app.axs\_Anh\_3\_before);

if length(app.info\_3) > 1

app.info\_3 = app.info\_3(1);

end

app.Lmax\_3 = 2^app.info\_3.BitDepth - 1;

if app.Lmax\_3 == 1

uialert(f, 'Ảnh vừa được chọn là ảnh nhị phân!', 'Lưu ý');

[numberofpixel, grayscale] = imhist(app.gray\_3);

bar(app.axs\_Hist\_3\_before, grayscale, numberofpixel, 0.5,"stacked", "black");

cla(app.axs\_Anh\_3\_after);

cla(app.axs\_Hist\_3\_after);

return;

end

% Xuất hist gốc và hist đã xử lý

[numberofpixel, grayscale] = imhist(app.gray\_3);

bar(app.axs\_Hist\_3\_before, grayscale, numberofpixel, 0.5,"stacked", "black");

gray\_3\_after = imadjust(app.gray\_3);

imshow(gray\_3\_after, 'Parent', app.axs\_Anh\_3\_after);[numberofpixel, grayscale] = imhist(gray\_3\_after);

bar(app.axs\_Hist\_3\_after, grayscale, numberofpixel, 0.5,"stacked", "blue");

end



f = app.axs\_Anh\_4\_after;

[filename, path] = uigetfile({'\*.png'; '\*.jpg'; '\*.tif'}, 'Chọn ảnh');

if isequal(filename, 0) || isequal(path, 0)

uialert(f, 'Chưa chọn ảnh', 'Lỗi đọc ảnh');

cla(app.axs\_Anh\_4\_before);

else

% Enable các thành phần cần thiết

app.bt\_Doc\_4.Text = 'Chọn ảnh khác';

% Đọc ảnh đã chọn và xuất ảnh xám

file = strcat(path, filename);

[app.img\_4, app.map\_4] = imread(file);

app.info\_4 = imfinfo(file);

app.colortype\_4 = app.info\_4.ColorType;

if app.colortype\_4 == "truecolor"

app.gray\_4 = rgb2gray(app.img\_4);

elseif app.colortype\_4 == "indexed"

app.gray\_4 = ind2gray(app.img\_4, app.map\_4);

else

app.gray\_4 = app.img\_4;

end

imshow(app.gray\_4, app.map\_4, 'Parent', app.axs\_Anh\_4\_before);

if length(app.info\_4) > 1

app.info\_4 = app.info\_4(1);

end

app.Lmax\_4 = 2^app.info\_4.BitDepth - 1;

if app.Lmax\_4 == 1

uialert(f, 'Ảnh vừa được chọn là ảnh nhị phân!', 'Lưu ý');

[numberofpixel, grayscale] = imhist(app.gray\_4);

bar(app.axs\_Hist\_4\_before, grayscale, numberofpixel, 0.5,"stacked", "black");

cla(app.axs\_Anh\_4\_after);

cla(app.axs\_Hist\_4\_after);

return;

end

% Xuất hist gốc và hist đã xử lý

[numberofpixel, grayscale] = imhist(app.gray\_4);

bar(app.axs\_Hist\_4\_before, grayscale, numberofpixel, 0.5,"stacked", "black");

gray\_4\_after = histeq(app.gray\_4);

imshow(gray\_4\_after, 'Parent', app.axs\_Anh\_4\_after);

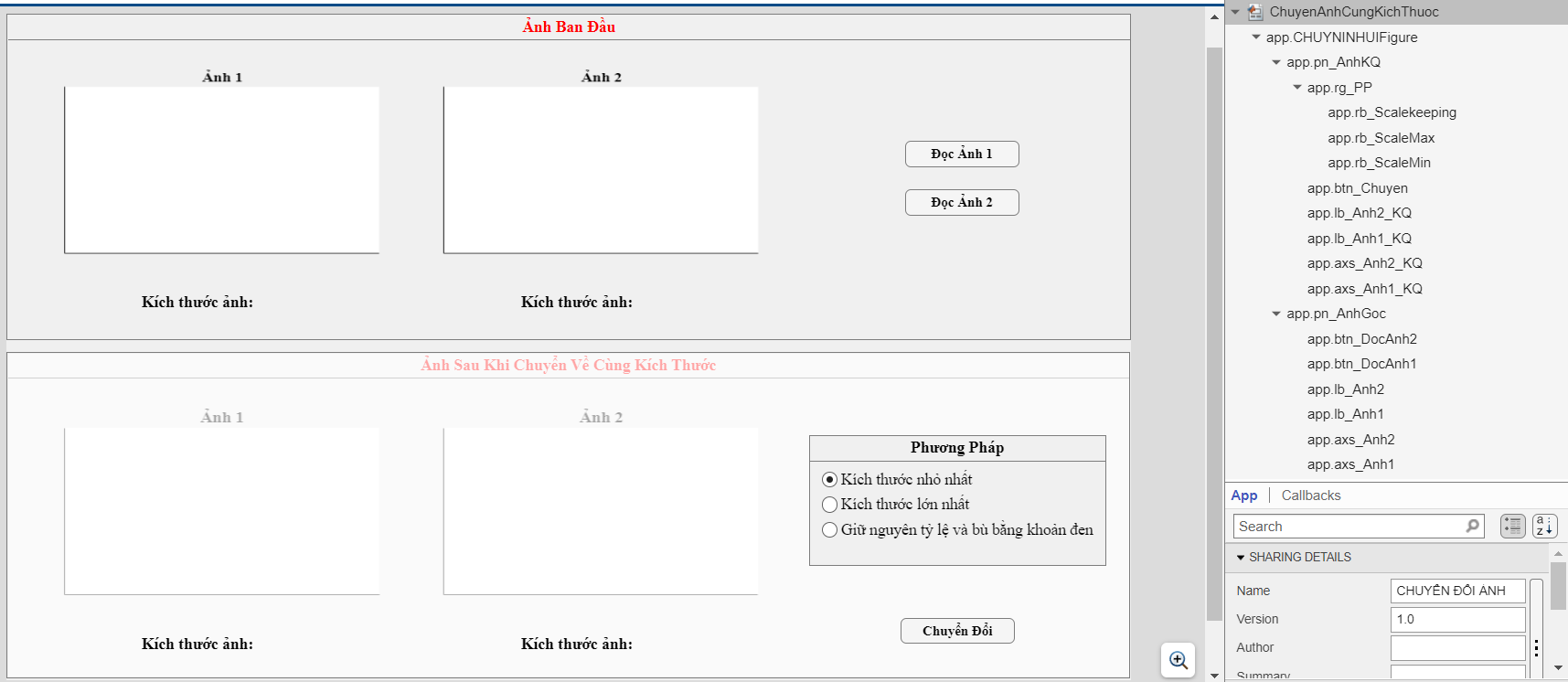
[numberofpixel, grayscale] = imhist(gray\_4\_after);

bar(app.axs\_Hist\_4\_after, grayscale, numberofpixel, 0.5,"stacked", "blue");

end

1. **Các phép toán số học trên ảnh *(có thể yêu cầu tự viết hàm hoặc dùng hàm dựng sẵn)***

* **Chuyển ảnh về cùng kích thước (chuyển thủ công hoặc dùng hàm imresize dùng kích thước ymax, xmax)**

****

Bỏ check thuộc tích Enable của đối tượng app.pn\_AnhKQ trong phần Interactivity của tab Inspector nhằm ngăn người dùng thao tác trên dữ liệu rỗng.

**Xây dựng giao diện:**

**•** **Panel** app.pn\_AnhGoc:

o **Axes**: app.axs\_Anh1, app.axs\_Anh2

o **Button**: app.btn\_DocAnh1, app.btn\_DocAnh2

o **Label:** app.lb\_Anh1, app.lb\_Anh2 (Hiển thị kích thước)

**• Panel** app.pn\_AnhKQ:

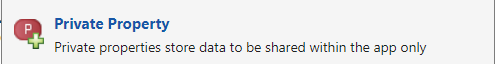
o **Axes:** app.axs\_Anh1\_KQ, app.axs\_Anh2\_KQ

o **Button:** app.btn\_Chuyen

o **Label**: app.lb\_Anh1\_KQ, app.lb\_Anh2\_KQ

o **Radio Button Group**: app.rg\_PP (Phương pháp) gồm có app.rb\_ScaleKeeping,

app.rb\_ScaleMin, app.rb\_ScaleMax

****

properties (Access = private)

AnhThuNhat\_Set = false

AnhThuHai\_Set = false

AnhCanResize1

MapAnhCanResize1

y1

x1

AnhCanResize2

MapAnhCanResize2

y2

x2

end

****

f = app.CHUYNINHUIFigure;

[filename, path] = uigetfile({'\*.png'; '\*.jpg'}, 'Chọn ảnh');

if isequal(filename, 0) || isequal(path, 0)

uialert(f, 'Chưa chọn ảnh', 'Lỗi đọc ảnh');

app.AnhThuNhat\_Set = false;

cla(app.axs\_Anh1);

else

app.AnhThuNhat\_Set = true;

file = strcat(path, filename);

[app.AnhCanResize1, app.MapAnhCanResize1] = imread(file);

imshow(app.AnhCanResize1, app.MapAnhCanResize1, 'Parent', app.axs\_Anh1);

app.y1 = size(app.AnhCanResize1, 1);

app.x1 = size(app.AnhCanResize1, 2);

app.lb\_Anh1.Text = "Kích thước ảnh: " + app.x1 + " x " + app.y1;

end

% Enable cho panel Chuyen doi

if app.AnhThuNhat\_Set == true && app.AnhThuHai\_Set == true

app.pn\_AnhKQ.Enable = "on";

else

app.pn\_AnhKQ.Enable = "off";

end

****

f = app.CHUYNINHUIFigure;

[filename, path] = uigetfile({'\*.png'; '\*.jpg'; '\*.tif'}, 'Chọn ảnh');

if isequal(filename, 0) || isequal(path, 0)

uialert(f, 'Chưa chọn ảnh', 'Lỗi đọc ảnh');

app.AnhThuHai\_Set = false;

cla(app.axs\_Anh2);

else

app.AnhThuHai\_Set = true;

file = strcat(path, filename);

[app.AnhCanResize2, app.MapAnhCanResize2] = imread(file);

imshow(app.AnhCanResize2, app.MapAnhCanResize2, 'Parent', app.axs\_Anh2);

app.y2 = size(app.AnhCanResize2, 1);

app.x2 = size(app.AnhCanResize2, 2);

app.lb\_Anh2.Text = "Kích thước ảnh: " + app.x2 + " x " + app.y2;

end

% Enable cho panel Chuyen doi

if app.AnhThuNhat\_Set == true && app.AnhThuHai\_Set == true

app.pn\_AnhKQ.Enable = "on";

else

app.pn\_AnhKQ.Enable = "off";

end



if app.rg\_PP.SelectedObject == app.rb\_ScaleMax

ymax = app.y1;

xmax = app.x1;

if app.y2 > ymax

ymax = app.y2;

end

if app.x2 > xmax

xmax = app.x2;

end

app.lb\_Anh1\_KQ.Text = "Kích thước ảnh: " + xmax + " x " + ymax;

app.lb\_Anh2\_KQ.Text = "Kích thước ảnh: " + xmax + " x " + ymax;

OI1 = imresize(app.AnhCanResize1, [ymax xmax]);

imshow(OI1, 'Parent', app.axs\_Anh1\_KQ);

OI2 = imresize(app.AnhCanResize2, [ymax xmax]);

imshow(OI2, 'Parent', app.axs\_Anh2\_KQ);

elseif app.rg\_PP.SelectedObject == app.rb\_ScaleMin

ymin = app.y1;

xmin = app.x1;

if app.y2 < ymin

ymin = app.y2;

end

if app.x2 < xmin

xmin = app.x2;

end

app.lb\_Anh1\_KQ.Text = "Kích thước ảnh: " + xmin + " x " + ymin;

app.lb\_Anh2\_KQ.Text = "Kích thước ảnh: " + xmin + " x " + ymin;

OI1 = imresize(app.AnhCanResize1, [ymin xmin]);

imshow(OI1, 'Parent', app.axs\_Anh1\_KQ);

OI2 = imresize(app.AnhCanResize2, [ymin xmin]);

imshow(OI2, 'Parent', app.axs\_Anh2\_KQ);

else

ymax = app.y1;

xmax = app.x1;

if app.y2 > ymax

ymax = app.y2;

end

if app.x2 > xmax

xmax = app.x2;

end

app.lb\_Anh1\_KQ.Text = "Kích thước ảnh: " + xmax + " x " + ymax;

app.lb\_Anh2\_KQ.Text = "Kích thước ảnh: " + xmax + " x " + ymax;

OI1 = padarray(app.AnhCanResize1, [(ymax - app.y1) (xmax - app.x1)], 0, 'post');

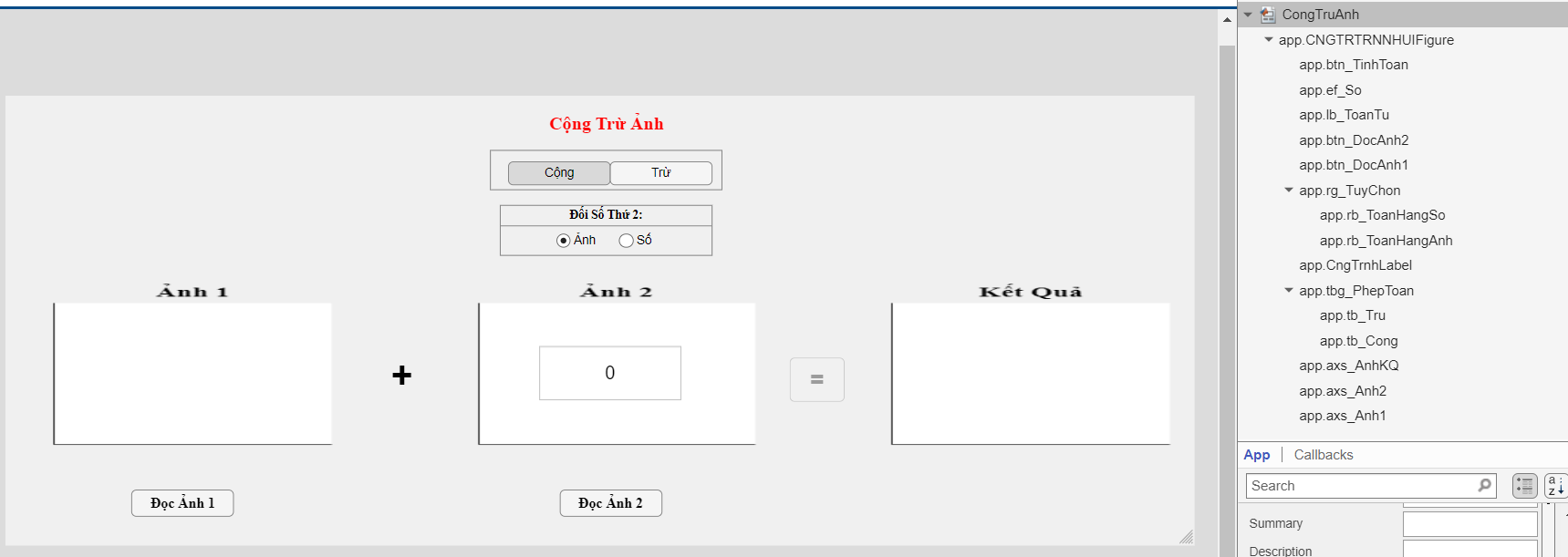
imshow(OI1, 'Parent', app.axs\_Anh1\_KQ);

OI2 = padarray(app.AnhCanResize2, [(ymax - app.y2) (xmax - app.x2)], 0, 'post');

imshow(OI2, 'Parent', app.axs\_Anh2\_KQ);

end

* **Cộng trừ ảnh**

Edit Field app.ef\_So nằm phía trên Axes app.axs\_Anh2 và bị uncheck thuộc tính Enable, Button btn\_TinhToan cũng bị uncheck thuộc tính Enable để tránh người dùng thao tác trên dữ liệu rỗng.

**Xây dựng giao diện:**

• **Toggle Button Group** app.tbg\_PhepToan gồm có app.tb\_Cong, app.tb\_Tru

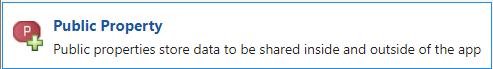
• **Radio Button Group** app.rg\_TuyChon gồm có app.rb\_ToanHangAnh, app.rb\_ToanHangSo

• **Button**: app.btn\_DocAnh1, app.btn\_DocAnh2, btn\_TinhToan (dấu =)

• **Label**: app.lb\_ToanTu (Hiển thị trực quan dấu + hoặc -)

• **Edit Field**: app.ef\_So

• **Axes**: app.axs\_Anh1, app.axs\_Anh2, app.axs\_AnhKQ

****

properties (Access = private)

A1\_Set = false

A2\_Set = false

AnhTrai

MapAnhTrai

AnhPhai

MapAnhPhai

end



selectedButton = app.tbg\_PhepToan.SelectedObject;

if selectedButton == app.tb\_Cong

app.axs\_Anh2.Visible = "on";

app.btn\_DocAnh2.Visible = "on";

app.rb\_ToanHangAnh.Value = 1;

app.ef\_So.Visible = "off";

app.rg\_TuyChon.Visible = "on";

app.lb\_ToanTu.Text = "+";

app.ef\_So.Value = 0;

cla(app.axs\_Anh2);

app.btn\_TinhToan.Enable = "off";

else

app.axs\_Anh2.Visible = "on";

app.btn\_DocAnh2.Visible = "on";

app.rb\_ToanHangAnh.Value = 1;

app.ef\_So.Visible = "off";

app.rg\_TuyChon.Visible = "on";

app.lb\_ToanTu.Text = "-";

app.ef\_So.Value = 0;

cla(app.axs\_Anh2);

app.btn\_TinhToan.Enable = "off";

end



selectedButton = app.rg\_TuyChon.SelectedObject;

if selectedButton == app.rb\_ToanHangAnh

app.axs\_Anh2.Visible = "on";

cla(app.axs\_Anh2);

app.A2\_Set = false;

app.btn\_TinhToan.Enable = "off";

app.btn\_DocAnh2.Visible = "on";

app.ef\_So.Visible = "off";

else

app.axs\_Anh2.Visible = "off";

cla(app.axs\_Anh2);

app.btn\_DocAnh2.Visible = "off";

app.ef\_So.Visible = "on";

if app.A1\_Set == true

app.btn\_TinhToan.Enable = "on";

end

end



f = app.axs\_Anh1;

[filename, path] = uigetfile({'\*.png'; '\*.jpg'; '\*.tif'}, 'Chọn ảnh');

if isequal(filename, 0) || isequal(path, 0)

uialert(f, 'Chưa chọn ảnh', 'Lỗi đọc ảnh');

app.A1\_Set = false;

cla(app.axs\_Anh1);

else

file = strcat(path, filename);

Lmax = 2^imfinfo(file).BitDepth - 1;

if Lmax == 1

uialert(f, 'Ảnh vừa chọn là ảnh nhị phân!', 'Cảnh báo');

return;

end

app.A1\_Set = true;

[app.AnhTrai, app.MapAnhTrai] = imread(file);

colortype = imfinfo(file).ColorType;

if colortype == "truecolor"

app.AnhTrai = rgb2gray(app.AnhTrai);

elseif colortype == "indexed"

app.AnhTrai = ind2gray(app.AnhTrai, app.MapAnhTrai);

end

imshow(app.AnhTrai, app.MapAnhTrai, 'Parent', app.axs\_Anh1);

if app.rg\_TuyChon.SelectedObject == app.rb\_ToanHangSo || app.A2\_Set == true app.btn\_TinhToan.Enable = "on";

end

end



f = app.axs\_Anh2;

[filename, path] = uigetfile({'\*.png'; '\*.jpg'; '\*.tif'}, 'Chọn ảnh');

if isequal(filename, 0) || isequal(path, 0)

uialert(f, 'Chưa chọn ảnh', 'Lỗi đọc ảnh');

app.A2\_Set = false;

cla(app.axs\_Anh2);

else

file = strcat(path, filename);

Lmax = 2^imfinfo(file).BitDepth - 1;

if Lmax == 1

uialert(f, 'Ảnh vừa chọn là ảnh nhị phân!', 'Cảnh báo');

return;

end

app.A2\_Set = true;

[app.AnhPhai, app.MapAnhPhai] = imread(file);

colortype = imfinfo(file).ColorType;

if colortype == "truecolor"

app.AnhPhai = rgb2gray(app.AnhPhai);

elseif colortype == "indexed"

app.AnhPhai = ind2gray(app.AnhPhai, app.MapAnhPhai);

end

imshow(app.AnhPhai, app.MapAnhPhai, 'Parent', app.axs\_Anh2);

end

% Enable cho nut =

if app.A1\_Set == true && app.A2\_Set == true

app.btn\_TinhToan.Enable = "on";

else

app.btn\_TinhToan.Enable = "off";

end



if app.tbg\_PhepToan.SelectedObject == app.tb\_Cong

if app.rg\_TuyChon.SelectedObject == app.rb\_ToanHangAnh

ymax = size(app.AnhTrai, 1);

xmax = size(app.AnhTrai, 2);

if size(app.AnhPhai, 1) > ymax

ymax = size(app.AnhPhai, 1);

end

if size(app.AnhPhai, 2) > xmax

xmax = size(app.AnhPhai, 2);

end

I1 = imresize(app.AnhTrai, [ymax xmax]);

I2 = imresize(app.AnhPhai, [ymax xmax]);

I = imadd(I1, I2, 'uint16');

imshow(I, [], 'Parent', app.axs\_AnhKQ);

else

c = app.ef\_So.Value;

KQ = imadd(app.AnhTrai, c);

imshow(KQ, [], 'Parent', app.axs\_AnhKQ);

end

else

if app.rg\_TuyChon.SelectedObject == app.rb\_ToanHangAnh

ymax = size(app.AnhTrai, 1);

xmax = size(app.AnhTrai, 2);

if size(app.AnhPhai, 1) > ymax

ymax = size(app.AnhPhai, 1);

end

if size(app.AnhPhai, 2) > xmax

xmax = size(app.AnhPhai, 2);

end

I1 = imresize(app.AnhTrai, [ymax xmax]);

I2 = imresize(app.AnhPhai, [ymax xmax]);

I = imsubtract(I1, I2);

imshow(I, [], 'Parent', app.axs\_AnhKQ);

else

c = app.ef\_So.Value;

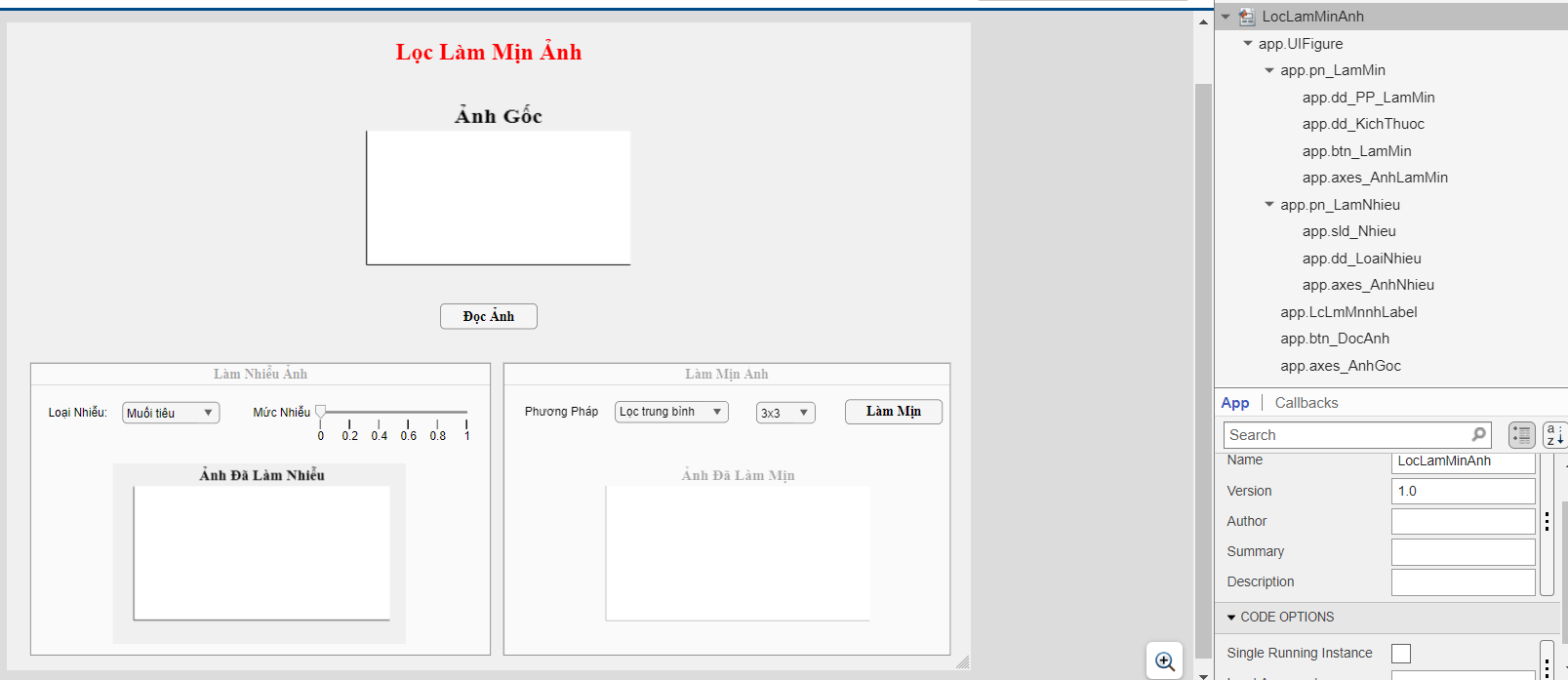
KQ = imsubtract(app.AnhTrai, c);

imshow(KQ, [], 'Parent', app.axs\_AnhKQ);

end

end

1. **Lọc làm mịn ảnh: mean, gaussian, median, order filter *(sử dụng các hàm lọc ảnh dựng sẵn)***

****

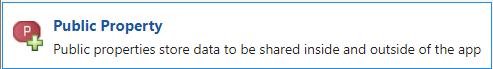
Bỏ check Enable cho app.pn\_LamNhieu và app.pn\_LamMin để tránh việc người dùng thao tác dữ liệu rỗng.

**Xây dựng giao diện:**

* + **Button:** app.btn\_DocAnh
  + **Axes**: app.axes\_AnhGoc

• **Panel** app.pn\_LamNhieu:

* + **Drop-down List**: app.dd\_LoaiNhieu
  + **Slider**: app.sld\_Nhieu
  + **Axes**: app.axes\_AnhNhieu
  + **Panel** app.pn\_LamMin:
  + **Drop-down List**: app.dd\_PP\_LamMin, app.dd\_KichThuoc
  + **Button**: app.btn\_LamMin
  + **Axes**: app.axes\_AnhLamMin

****

properties (Access = private)

img

map

info

colortype

img\_gray

img\_noise

img\_lm

end

****

f = app.UIFigure;

[filename, path] = uigetfile({'\*.png'; '\*.jpg'}, 'Chọn ảnh');

if isequal(filename, 0) || isequal(path, 0)

uialert(f, "Chưa chọn ảnh nào", "Lỗi đọc ảnh");

else

% ---

app.pn\_LamMin.Enable = "off";

app.pn\_LamNhieu.Enable = "on";

app.btn\_DocAnh.Text = "Đọc ảnh khác";

app.sld\_Nhieu.Value = 0;

app.dd\_KichThuoc.Value = "3x3";

app.dd\_LoaiNhieu.Value = "Muối tiêu";

app.dd\_PP\_LamMin.Value = "Lọc trung bình";

% ---

file = strcat(path, filename);

app.info = imfinfo(file);

Lmax = 2^app.info.BitDepth - 1;

if Lmax == 1

uialert(f, "Ảnh vừa chọn là ảnh nhị phân", "Lỗi đọc ảnh");

return;

end

[app.img, app.map] = imread(file);

app.colortype = app.info.ColorType;

if app.colortype == "truecolor"

app.img\_gray = rgb2gray(app.img);

elseif app.colortype == "indexed"

app.img\_gray = ind2gray(app.img, app.map);

else

app.img\_gray = app.img;

end

imshow(app.img\_gray, 'Parent', app.axes\_AnhGoc);

imshow(app.img\_gray, 'Parent', app.axes\_AnhNhieu);

cla(app.axes\_AnhLamMin);

end

****

changingValue = event.Value;

if app.dd\_LoaiNhieu.Value == "Muối tiêu"

app.img\_noise = imnoise(app.img\_gray, "salt & pepper",changingValue);

app.axes\_AnhNhieu.Title.String = "Ảnh nhiễu muối tiêu";

else

app.img\_noise = imnoise(app.img\_gray, "gaussian", changingValue);

app.axes\_AnhNhieu.Title.String = "Ảnh nhiễu Gaussian";

end

imshow(app.img\_noise, 'Parent', app.axes\_AnhNhieu);

imshow(app.img\_noise, 'Parent', app.axes\_AnhLamMin);

app.pn\_LamMin.Enable = "on";

app.axes\_AnhLamMin.Title.String = "Ảnh cần được làm mịn";

****

i = app.img\_noise;

if app.dd\_PP\_LamMin.Value == "Lọc trung bình"

if app.dd\_KichThuoc.Value == "3x3"

bl = fspecial("average", [3, 3]);

elseif app.dd\_KichThuoc.Value == "5x5"

bl = fspecial("average", [5, 5]);

elseif app.dd\_KichThuoc.Value == "7x7"

bl = fspecial("average", [7, 7]);

elseif app.dd\_KichThuoc.Value == "11x11"

bl = fspecial("average", [11, 11]);

end

app.img\_lm = filter2(bl, i);

imshow(app.img\_lm/255, 'Parent', app.axes\_AnhLamMin);

elseif app.dd\_PP\_LamMin.Value == "Lọc trung vị"

if app.dd\_KichThuoc.Value == "3x3"

app.img\_lm = medfilt2(i, [3, 3]);

elseif app.dd\_KichThuoc.Value == "5x5"

app.img\_lm = medfilt2(i, [5, 5]);

elseif app.dd\_KichThuoc.Value == "7x7"

app.img\_lm = medfilt2(i, [7, 7]);

elseif app.dd\_KichThuoc.Value == "11x11"

app.img\_lm = medfilt2(i, [11, 11]);

end

imshow(app.img\_lm, 'Parent', app.axes\_AnhLamMin);

elseif app.dd\_PP\_LamMin.Value == "Lọc Gaussian"

if app.dd\_KichThuoc.Value == "0.5"

app.img\_lm = imgaussfilt(i, 0.5);

elseif app.dd\_KichThuoc.Value == "1"

app.img\_lm = imgaussfilt(i, 1);

elseif app.dd\_KichThuoc.Value == "2"

app.img\_lm = imgaussfilt(i, 2);

elseif app.dd\_KichThuoc.Value == "4"

app.img\_lm = imgaussfilt(i, 4);

elseif app.dd\_KichThuoc.Value == "8"

app.img\_lm = imgaussfilt(i, 8);

end

imshow(app.img\_lm, 'Parent', app.axes\_AnhLamMin);

elseif app.dd\_PP\_LamMin.Value == "Lọc Max"

if app.dd\_KichThuoc.Value == "3x3"

app.img\_lm = ordfilt2(i, 9, ones(3, 3));

elseif app.dd\_KichThuoc.Value == "5x5"

app.img\_lm = ordfilt2(i, 25, ones(5, 5));

elseif app.dd\_KichThuoc.Value == "7x7"

app.img\_lm = ordfilt2(i, 49, ones(7, 7));

elseif app.dd\_KichThuoc.Value == "11x11"

app.img\_lm = ordfilt2(i, 121, ones(11, 11));

end

imshow(app.img\_lm, 'Parent', app.axes\_AnhLamMin);

elseif app.dd\_PP\_LamMin.Value == "Lọc Min"

if app.dd\_KichThuoc.Value == "3x3"

app.img\_lm = ordfilt2(i, 1, ones(3, 3));

elseif app.dd\_KichThuoc.Value == "5x5"

app.img\_lm = ordfilt2(i, 1, ones(5, 5));

elseif app.dd\_KichThuoc.Value == "7x7"

app.img\_lm = ordfilt2(i, 1, ones(7, 7));

elseif app.dd\_KichThuoc.Value == "11x11"

app.img\_lm = ordfilt2(i, 1, ones(11, 11));

end

imshow(app.img\_lm, 'Parent', app.axes\_AnhLamMin);

end

app.axes\_AnhLamMin.Title.String = "Ảnh đã được làm mịn";

****

value = app.dd\_PP\_LamMin.Value;

if value == "Lọc Gaussian"

app.dd\_KichThuoc.Items = ["0.5", "1", "2", "4", "8"];

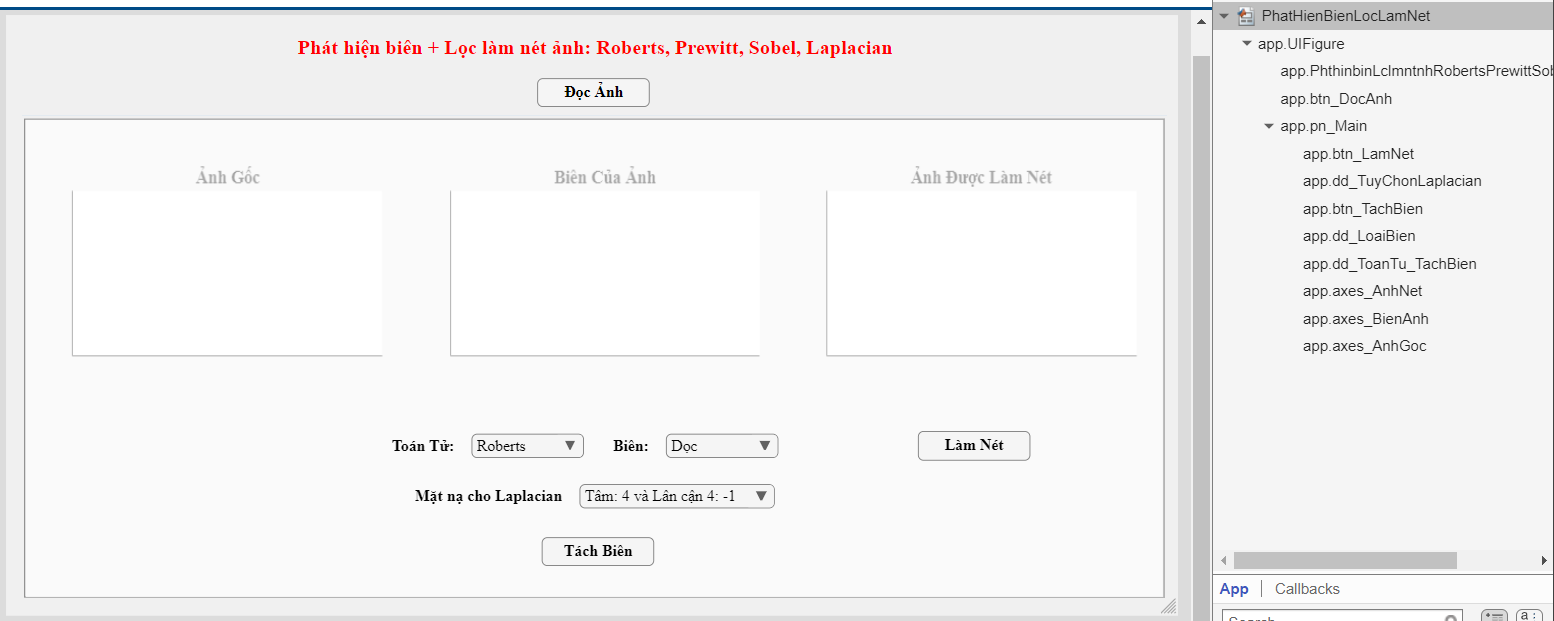
else

app.dd\_KichThuoc.Items = ["3x3", "5x5", "7x7", "11x11"];

end

1. **Phát hiện biên + lọc làm nét ảnh: Robert, Prewitt, Sobel, Laplacian *(chỉ sử dụng các hàm lọc ảnh dựng sẵn)***

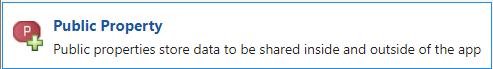
* **Hiển thí các kết quả trung gian (ảnh biên ngang, biên dọc) nếu có**
* **Tùy biến bộ lọc Laplacian với các bản mở rộng**

****

Bỏ check thuộc tính Enable của app.pn\_Main để tránh việc người dùng thao tác dữ liệu rỗng.

**Xây dựng giao diện:**

* **Button:** app.btn\_DocAnh
* **Panel** app.pn\_Main:
* **Axes**: app.axes\_AnhGoc, app.axes\_BienAnh, app.axes\_AnhNet
* **Drop-down List**: app.dd\_ToanTu\_TachBien, app.dd\_LoaiBien, app.dd\_TuyChonLaplacian
* **Button**: app.btn\_TachBien, app.btn\_LamNet

****

properties (Access = private)

img

map

info

colortype

img\_gray

img\_edge

img\_ln

% Roberts

Gx\_Rb = [0, 1; -1, 0;];

Gy\_Rb = [1, 0; 0, -1;];

% Prewitt

Gx\_Prw = [-1, 0, 1; -1, 0, 1; -1, 0, 1;];

Gy\_Prw = [-1, -1, -1; 0, 0, 0; 1, 1, 1;];

% Sobel

Gx\_Sb = [-1, 0, 1; -2, 0, 2; -1, 0, 1;];

Gy\_Sb = [-1, -2, -1; 0, 0, 0; 1, 2, 1;];

% Laplace

L\_4 = [0, -1, 0; -1, 4, -1; 0, -1, 0;];

L\_8 = [-1, -1, -1; -1, 8, -1; -1, -1, -1;];

end



f = app.UIFigure;

[filename, path] = uigetfile({'\*.png'; '\*.jpg'}, 'Chọn ảnh');

if isequal(filename, 0) || isequal(path, 0)

uialert(f, "Chưa chọn ảnh nào", "Lỗi đọc ảnh");

else

% ---

app.pn\_Main.Enable = "on";

app.btn\_DocAnh.Text = "Đọc ảnh khác";

app.dd\_LoaiBien.Items = ["Dọc", "Ngang", "Toàn biên"];

app.dd\_LoaiBien.Value = "Dọc";

app.dd\_ToanTu\_TachBien.Value = "Roberts";

app.dd\_TuyChonLaplacian.Enable = "off";

app.axes\_AnhNet.Title.String = "Ảnh cần được làm nét";

% ---

file = strcat(path, filename);

app.info = imfinfo(file);

Lmax = 2^app.info.BitDepth - 1;

if Lmax == 1

uialert(f, "Ảnh vừa chọn là ảnh nhị phân", "Lưu ý");

return;

end

[app.img, app.map] = imread(file);

app.colortype = app.info.ColorType;

if app.colortype == "truecolor"

app.img\_gray = rgb2gray(app.img);

elseif app.colortype == "indexed"

app.img\_gray = ind2gray(app.img, app.map);

else

app.img\_gray = app.img;

end

imshow(app.img\_gray, 'Parent', app.axes\_AnhNet);

imshow(app.img\_gray, 'Parent', app.axes\_AnhGoc);

cla(app.axes\_BienAnh);

end



i = app.img\_gray;

if app.dd\_ToanTu\_TachBien.Value == "Roberts"

% Tách biên Robert

edge\_x = filter2(app.Gx\_Rb, i);

edge\_y = filter2(app.Gy\_Rb, i);

app.img\_edge = sqrt(edge\_x.^2 + edge\_y.^2);

app.img\_ln = imadd(imfilter(i, app.Gx\_Rb), i);

app.img\_ln = imadd(imfilter(i, app.Gy\_Rb), i);

elseif app.dd\_ToanTu\_TachBien.Value == "Prewitt"

%fs = fspecial('prewitt');

%fil = imfilter(i, fs);

%app.img\_ln = imadd(fil , i);

edge\_x = filter2(app.Gx\_Prw, i);

edge\_y = filter2(app.Gy\_Prw, i);

app.img\_edge = sqrt(edge\_x.^2 + edge\_y.^2);

app.img\_ln = imadd(imfilter(i, app.Gx\_Prw), i);

app.img\_ln = imadd(imfilter(i, app.Gy\_Prw), i);

elseif app.dd\_ToanTu\_TachBien.Value == "Sobel"

%fs = fspecial('sobel');

%fil = imfilter(i, fs);

%app.img\_ln = imadd(fil , i);

edge\_x = filter2(app.Gx\_Sb, i);

edge\_y = filter2(app.Gy\_Sb, i);

app.img\_edge = sqrt(edge\_x.^2 + edge\_y.^2);

app.img\_ln = imadd(imfilter(i, app.Gx\_Sb), i);

app.img\_ln = imadd(imfilter(i, app.Gy\_Sb), i);

elseif app.dd\_ToanTu\_TachBien.Value == "Laplace"

%fs = fspecial('laplacian', 0);

%fil = imfilter(i, fs);

%app.img\_ln = imadd(fil, i);

if app.dd\_TuyChonLaplacian.Value == "Tâm: 4 và Lân cận 4: -1"

app.img\_edge = filter2(app.L\_4, i);

app.img\_ln = imadd(i, imfilter(i, app.L\_4));

elseif app.dd\_TuyChonLaplacian.Value == "Tâm: 8 và Lân cận 8: -1"

app.img\_edge = filter2(app.L\_8, i);

app.img\_ln = imadd(i, imfilter(i, app.L\_8));

end

end

if app.dd\_LoaiBien.Value == "Dọc"

imshow(edge\_x/255, 'Parent', app.axes\_BienAnh);

elseif app.dd\_LoaiBien.Value == "Ngang"

imshow(edge\_y/255, 'Parent', app.axes\_BienAnh);

else

imshow(app.img\_edge/255, 'Parent', app.axes\_BienAnh);

end

app.btn\_LamNet.Enable = "on";



imshow(app.img\_ln, app.map, 'Parent', app.axes\_AnhNet);

app.axes\_AnhNet.Title.String = "Ảnh đã được làm nét";



value = app.dd\_ToanTu\_TachBien.Value;

if value == "Laplace"

app.dd\_LoaiBien.Items = "Toàn biên";

app.dd\_TuyChonLaplacian.Enable = "on";

else

app.dd\_LoaiBien.Items = ["Dọc", "Ngang", "Toàn biên"];

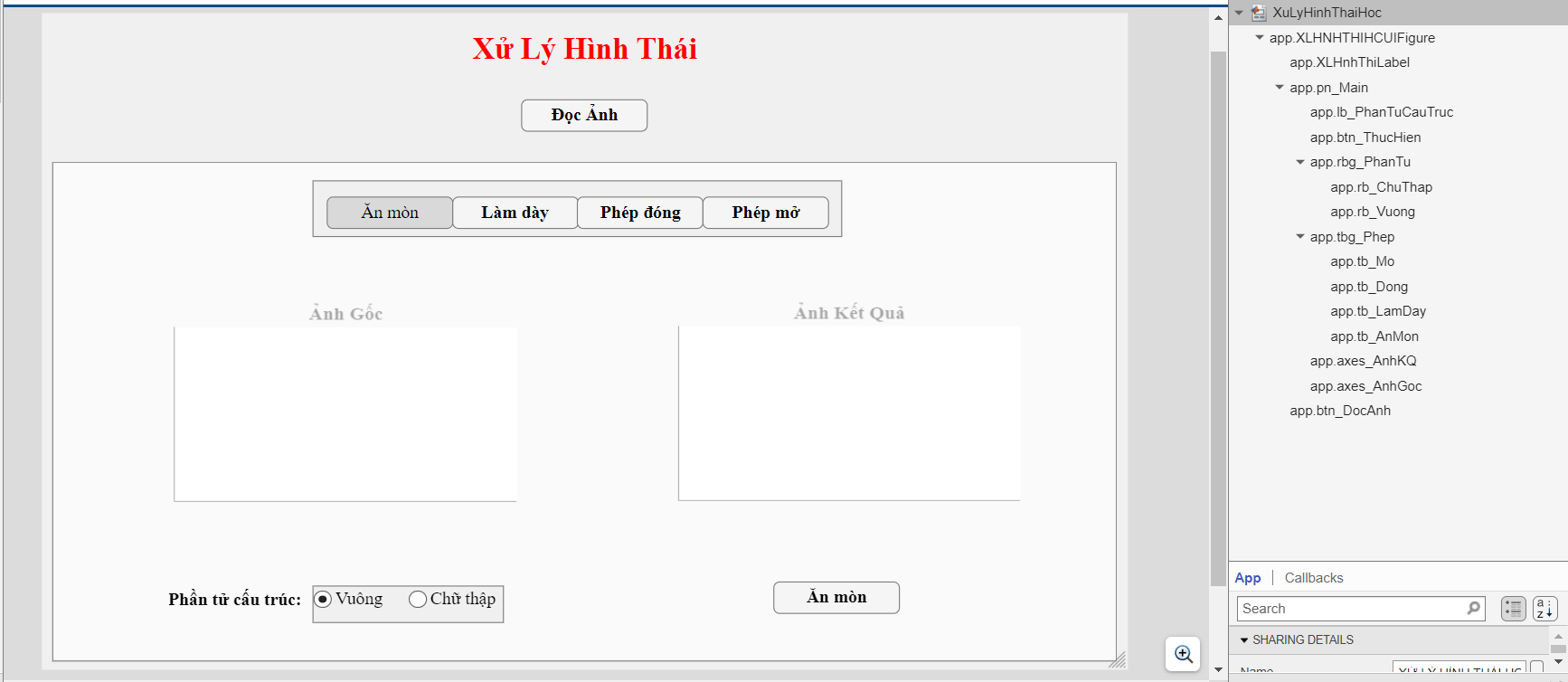
app.dd\_LoaiBien.Value = "Dọc";

app.dd\_TuyChonLaplacian.Enable = "off";

end

1. **Xử lý hình thái**

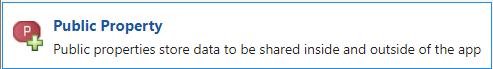
* **Ăn mòn, làm dày ảnh**
* **Đóng ảnh, mở ảnh**
* **Tùy biến phần tử cấu trúc chữ thập hoặc hình vuông.**

****

Bỏ check thuộc tính Enable của app.pn\_Main để tránh người dùng thao tác dữ liệu rỗng.

**Xây dựng giao diện:**

* **Button**: app.btn\_DocAnh
* **Panel** app.pn\_Main:
  + **Toggle Button Group** app.tbg\_Phep gồm có app.tb\_AnMon, app.tb\_LamDay, app.tb\_Dong, app.tb\_Mo
  + **Axes**: app.axes\_AnhGoc, app.axes\_AnhKQ
  + **Radio Button Group** app.rbg\_PhanTu gồm có app.rb\_Vuong, app.rb\_ChuThap
  + **Button**: app.btn\_ThucHien

****

properties (Access = private)

img

map

colortype

img\_kq

end



[filename, path] = uigetfile({'\*.png'; '\*.jpg'}, 'Chọn ảnh');

if ~isequal(filename, 0) && ~isequal(path, 0)

app.btn\_DocAnh.Text = "Đọc ảnh khác";

app.pn\_Main.Enable = "on";

app.tbg\_Phep.SelectedObject = app.tb\_AnMon;

app.btn\_ThucHien.Text = "Ăn mòn";

app.rbg\_PhanTu.SelectedObject = app.rb\_Vuong;

cla(app.axes\_AnhKQ);

file = strcat(path, filename);

[app.img, app.map] = imread(file);

app.colortype = imfinfo(file).ColorType;

if app.colortype == "truecolor"

app.img = rgb2gray(app.img);

app.img = imbinarize(app.img);

elseif app.colortype == "indexed"

app.img = ind2gray(app.img, app.map);

app.img = imbinarize(app.img);

end

imshow(app.img, app.map, 'Parent', app.axes\_AnhGoc);

end



if app.rbg\_PhanTu.SelectedObject == app.rb\_Vuong

ptct = ones(3, 3);

else

ptct = strel([0, 1, 0; 1, 1, 1; 0, 1, 0]);

end

if app.tbg\_Phep.SelectedObject == app.tb\_AnMon

app.img\_kq = imerode(app.img, ptct);

elseif app.tbg\_Phep.SelectedObject == app.tb\_LamDay

app.img\_kq = imdilate(app.img, ptct);

elseif app.tbg\_Phep.SelectedObject == app.tb\_Dong

app.img\_kq = imdilate(app.img, ptct);

app.img\_kq = imerode(app.img\_kq, ptct);

elseif app.tbg\_Phep.SelectedObject == app.tb\_Mo

app.img\_kq = imerode(app.img, ptct);

app.img\_kq = imdilate(app.img\_kq, ptct);

end

imshow(app.img\_kq, "Parent", app.axes\_AnhKQ);



selectedButton = app.tbg\_Phep.SelectedObject;

if selectedButton == app.tb\_AnMon

app.btn\_ThucHien.Text = "Ăn mòn";

elseif selectedButton == app.tb\_LamDay

app.btn\_ThucHien.Text = "Làm dày";

else

app.btn\_ThucHien.Text = "Thực hiện";

end

1. **Sử dụng các components trong component library của MATLAB App Designer để tuỳ biến các thông số, bộ lọc; thiết kế giao diện đảm bảo tính tương tác tốt với người dùng.**

**Axes:** Hiển thị ảnh (gốc, xử lý,... ):

imshow(Ảnh, Map của ảnh, "Parent", Axes chứa ảnh);

**Hiển thị ảnh phân ngưỡng đơn:**

imshow(Ảnh > giá trị pn, Map của ảnh, "Parent", Axes chứa ảnh);

**Hiển thị ảnh phân ngưỡng kép:**

imshow(Ảnh > giá trị pn trái & Ảnh < giá trị pn phải, Map của ảnh, "Parent", Axes chứa ảnh);

**Hiển thị tổ chức đồ:**

[numberofpixel, grayscale] = imhist(Ảnh);

bar(Axes chứa tổ chức đồ, grayscale, numberofpixel, 0.5, "stacked", "black");

- Sự kiện Button Down: thực hiện đoạn lệnh đã nhập khi người dùng nhấp chuột vào khu vực của Axes.

**Button**: Nút ấn

- Sự kiện Button Pushed: thực hiện đoạn lệnh đã nhập khi người dùng ấn nút xong.

**Check Box**: Ô đánh dấu ✓

- Sự kiện Value Changed: thực hiện đoạn lệnh đã nhập khi trạng thái đánh dấu bị người dùng thay đổi ( →  hoặc  → ).

**Drop-down**: Danh sách xổ

* Sự kiện Value Changed: thực hiện đoạn lệnh đã nhập khi người dùng chọn xong một đối tượng

trong danh sách xổ xuống khác với đối tượng trước đã chọn.

Sự kiện Drop-down Opening: thực hiện đoạn lệnh đã nhập ngay khi người dùng vừa xổ danh sách.

Thường dùng cho việc khởi tạo danh sách xổ một cách chủ động trong cửa sổ soạn thảo lệnh.

Khởi tạo danh sách bằng thuộc tính .Items, ví dụ:

app.dd\_PhepToan.Items = ["Cộng", "Trừ", "Nhân", "Chia"];

**Edit Field (Numeric):** Ô nhập số

Sự kiện Value Changed: thực hiện đoạn lệnh đã nhập khi người dùng thay đổi giá trị trong ô nhập.

**(Radio) Button Group**: Nhóm các nút lựa chọn

Sự kiện Selection Changed: thực hiện đoạn lệnh đã nhập khi người dùng chọn một Radio Button

khác với Radio Button đã chọn trước đó.

Sự kiện Button Down: thực hiện đoạn lệnh đã nhập khi người dùng nhấp chuột vào khu vực của

Button Group.

**Slider**: Thanh trượt chọn giá trị

Sự kiện Value Changing: thực hiện đoạn lệnh đã nhập ngay khi người dùng còn đang kéo trượt.

Sự kiện Value Changed: thực hiện đoạn lệnh đã nhập khi người dùng kéo trượt xong và thả chuột.

**Toggle Button Group**: Nhóm các nút chuyển đổi (tương tự như Radio Button Group)

Sự kiện Selection Changed: thực hiện đoạn lệnh đã nhập khi người dùng chọn một Toggle Button

khác với Toggle Button đã chọn trước đó.

Thích hợp để thiết lập giao diện chuyển đổi các chế độ hay chức năng