function varargout = mainfig(varargin)

% MAINFIG MATLAB code for mainfig.fig

% MAINFIG, by itself, creates a new MAINFIG or raises the existing

% singleton\*.

%

% H = MAINFIG returns the handle to a new MAINFIG or the handle to

% the existing singleton\*.

%

% MAINFIG('CALLBACK',hObject,eventData,handles,...) calls the local

% function named CALLBACK in MAINFIG.M with the given input arguments.

%

% MAINFIG('Property','Value',...) creates a new MAINFIG or raises the

% existing singleton\*. Starting from the left, property value pairs are

% applied to the GUI before mainfig\_OpeningFcn gets called. An

% unrecognized property name or invalid value makes property application

% stop. All inputs are passed to mainfig\_OpeningFcn via varargin.

%

% \*See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one

% instance to run (singleton)".

%

% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help mainfig

% Last Modified by GUIDE v2.5 04-Jun-2017 12:36:53

% Begin initialization code - DO NOT EDIT

gui\_Singleton = 1;

gui\_State = struct('gui\_Name', mfilename, ...

'gui\_Singleton', gui\_Singleton, ...

'gui\_OpeningFcn', @mainfig\_OpeningFcn, ...

'gui\_OutputFcn', @mainfig\_OutputFcn, ...

'gui\_LayoutFcn', [] , ...

'gui\_Callback', []);

if nargin && ischar(varargin{1})

gui\_State.gui\_Callback = str2func(varargin{1});

end

if nargout

[varargout{1:nargout}] = gui\_mainfcn(gui\_State, varargin{:});

else

gui\_mainfcn(gui\_State, varargin{:});

end

% End initialization code - DO NOT EDIT

% --- Executes just before mainfig is made visible.

function mainfig\_OpeningFcn(hObject, eventdata, handles, varargin)

% This function has no output args, see OutputFcn.

% hObject handle to figure

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

% varargin command line arguments to mainfig (see VARARGIN)

% Choose default command line output for mainfig

handles.output = hObject;

% Update handles structure

guidata(hObject, handles);

% UIWAIT makes mainfig wait for user response (see UIRESUME)

% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.

function varargout = mainfig\_OutputFcn(hObject, eventdata, handles)

% varargout cell array for returning output args (see VARARGOUT);

% hObject handle to figure

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure

varargout{1} = handles.output;

% --- Executes on button press in Input Image.

function pushbutton1\_Callback(hObject, eventdata, handles)

% hObject handle to pushbutton1 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

[filename, pathname] = uigetfile({'\*.png';'\*.bmp';'\*.jpg';'\*.\*'}, 'Pick an Image File');

S1 = imread([pathname,filename]);

% S1 =imresize(S1,[256 256]);

axes(handles.axes5);

imshow(S1);

handles.S1 = S1;

guidata(hObject, handles);

%%

% --- Executes on button press in morphological operation.

function pushbutton2\_Callback(hObject, eventdata, handles)

% hObject handle to pushbutton2 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

I= handles.S1;

R=I(:,:,1);

G=I(:,:,2);

B=I(:,:,3);

[m n]=size(R);

for i=1:m

for j=1:n

if R(i,j)>200

R(i,j)=0;

G(i,j)=0;

B(i,j)=0;

end

end

end

v=cat(3,R,G,B);

bw=edge(rgb2gray(v),'canny');

se = strel('disk',1);

bw2 = imdilate(bw,se);

bw1=imfill(bw2,'holes');

bw1=bwareaopen(bw1,100);

stats=regionprops(bw1,'all');

n1=cat(1,stats.BoundingBox);

[rr1 rr2]=size(n1);

togglefig('gray scale image');

% imshow(S1);

imshow(bw1)

togglefig('output image');

% imshow(S1);

imshow(I)

hold on

for i=1:rr1

rectangle('Position',n1(i,:),'edgecolor','red');

end

for i=1:rr1

RR=imcrop(R,n1(i,:));

GG=imcrop(G,n1(i,:));

BB=imcrop(B,n1(i,:));

GLCM2 = graycomatrix(RR,'Offset',[2 0;0 2]);

stats1 = graycoprops(GLCM2,{'contrast','homogeneity','Energy','Correlation'});

ss1=cat(2,stats1.Contrast,stats1.Homogeneity,stats1.Energy,stats1.Correlation);

GLCM2 = graycomatrix(GG,'Offset',[2 0;0 2]);

stats2 = graycoprops(GLCM2,{'contrast','homogeneity','Energy','Correlation'});

ss2=cat(2,stats2.Contrast,stats2.Homogeneity,stats2.Energy,stats2.Correlation);

GLCM2 = graycomatrix(BB,'Offset',[2 0;0 2]);

stats3 = graycoprops(GLCM2,{'contrast','homogeneity','Energy','Correlation'});

ss3=cat(2,stats3.Contrast,stats3.Homogeneity,stats3.Energy,stats3.Correlation);

%rectangle('Position',n1(i,:),'edgecolor','red');

end

GLCM2 = graycomatrix(bw1,'Offset',[2 0;0 2]);

stats3 = graycoprops(GLCM2,{'contrast','homogeneity','Energy','Correlation'});

feat=cat(2,stats3.Contrast,stats3.Homogeneity,stats3.Energy,stats3.Correlation);

%%

function edit1\_Callback(hObject, eventdata, handles)

% hObject handle to edit1 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit1 as text

% str2double(get(hObject,'String')) returns contents of edit1 as a double

% --- Executes during object creation, after setting all properties.

function edit1\_CreateFcn(hObject, eventdata, handles)

% hObject handle to edit1 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.

% See ISPC and COMPUTER.

if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))

set(hObject,'BackgroundColor','white');

end

%%

% --- Executes on button press in combined algorithm.

function pushbutton3\_Callback(hObject, eventdata, handles)

% hObject handle to pushbutton3 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

I= handles.S1;

targetImage = I;

%%

cellMask = segmentImageFcn(targetImage);

cellMask = bwareaopen(cellMask,100);

CM=cellMask;

togglefig('Cell Mask',true)

imshow(CM);

title('Cell mask','fontsize',14);

%% Watershed Segmentation

grayscale = rgb2gray(targetImage);

imshow(grayscale)

wsImg = watershed(grayscale);

showMaskAsOverlay(0.8, wsImg==0, 'g')

%% Improving that result:

% Consider the image as a "topography"

% togglefig('Cell Mask',true)

% clf

% ax3(1) = subplot(1,2,1);

ds = 3;

surf(im2double(grayscale(1:ds:end,1:ds:end)));

% shading interp;

% rotate3d on

set(gca,'view',[0 90],...

'xlim',[0 100],'ylim',[0 100],'zlim',[0.3 1])

% title('Original Grayscale','fontsize',12)

% drawnow

% IMHMIN:

newGrayscale = imhmin(grayscale,13);

% ax3(2) = subplot(1,2,2);

surf(im2double(newGrayscale(1:ds:end,1:ds:end)));

% shading interp

% rotate3d on

% set(gca,'view',[0 90],...

% 'xlim',[0 100],'ylim',[0 100],'zlim',[0.3 1])

% title('Grayscale with Suppressed Minima','fontsize',12)

% linkprop(ax3,'view');

% colormap(flipud(parula));

%% Better segementation

%togglefig('Exploration')

newGrayscale = imhmin(grayscale,13);

wsImg = watershed(newGrayscale);

showMaskAsOverlay(1,wsImg == 0, 'g');

%title('Better...','fontsize',14);

%% So now we can use this calculation to improve our segmentation mask

togglefig('Cell Mask',true)

cellMask = segmentImageFcn(targetImage);

cellMask = bwareaopen(cellMask,30);

imshow(cellMask);

%% Impose watershed lines

% Just like we did with the edges calculated earlier, we can break our

% regions along watershed lines

togglefig('Cell Mask',true)

wsEdges = wsImg == 0;

wsEdges = bwareaopen(wsEdges,200,8);

cellMask(wsEdges) = 0;

imshow(cellMask);

% circleFinder(grayscale)

%%

detectCircles=@(x)imfindcircles(x,[20,35],'Sensitivity',0.89,'EdgeThreshold',0.04,'Method','TwoStage','ObjectPolarity','Dark');

detectCircles = @(x) imfindcircles(x,[20 35], ...

'Sensitivity',0.89, ...

'EdgeThreshold',0.04, ...

'Method','TwoStage', ...

'ObjectPolarity','Dark');

[centers, radii, metric] = detectCircles(grayscale);

togglefig('Target Image',true)

imshow(targetImage)

viscircles(centers,radii,'edgecolor','b')

title(sprintf('%i Cells Detected',numel(radii)),'fontsize',14);

%%

infectionThreshold=135;

%%

[centers,radii] = detectCircles(grayscale);

isInfected = false(numel(radii),1);

nCells = numel(isInfected);

%

% Creating a "mesh" can be useful:

x = 1:size(grayscale,2);

y = 1:size(grayscale,1);

[xx,yy] = meshgrid(x,y);

% xx(1:5,1:5),yy(1:5,1:5)

togglefig('Result mask',true);

imshow(grayscale)

infectionMask = false(size(grayscale));

for ii = 1:numel(radii)

mask = hypot(xx - centers(ii,1), yy - centers(ii,2)) <= radii(ii);

currentCellImage = grayscale;

currentCellImage(~mask) = 0;

infection = ...

currentCellImage > 0 & currentCellImage < infectionThreshold;

% OPTIONALLY set a range for the calculated regionproperties of

% infections:

% infection = bwpropfilt(infection,'Eccentricity',[0,0.925]);

infectionMask = infectionMask | infection;

isInfected(ii) = any(infection(:));

if isInfected(ii)

showMaskAsOverlay(0.3,mask,'g',[],false);

end

end

showMaskAsOverlay(0.5,infectionMask,'r',[],false)

title(sprintf('%i of %i (%0.1f%%) Infected',...

sum(isInfected),numel(isInfected),...

100\*sum(isInfected)/numel(isInfected)),...

'fontsize',14,'color','r');

% --- Executes on button press in accuracy of morphological operation.

function pushbutton4\_Callback(hObject, eventdata, handles)

% hObject handle to pushbutton4 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

% I=imread('D:\Right.jpg');

load('GroundTruth');

% TRIANING PART

mm='C:\Users\vysha\Desktop\malaria detection\Train\'; % TO READ TRAINING SAMPLES

DatabasePath = mm;

TIFfile = dir(fullfile(DatabasePath, '\\*.jpg'));

for index2 = 1 : length(TIFfile)

fileName = fullfile(DatabasePath, TIFfile(index2).name);

I=imread(fileName);

R=I(:,:,1);

G=I(:,:,2);

B=I(:,:,3);

[m n]=size(R);

for i=1:m

for j=1:n

if R(i,j)>200

R(i,j)=0;

G(i,j)=0;

B(i,j)=0;

end

end

end

v=cat(3,R,G,B);

bw=edge(rgb2gray(v),'canny');

se = strel('disk',1);

bw2 = imdilate(bw,se);

bw1=imfill(bw2,'holes');

bw1=bwareaopen(bw1,100);

stats=regionprops(bw1,'all');

n1=cat(1,stats.BoundingBox);

% [rr1 rr2]=size(n1);

% figure, imshow(I)

% hold on

% for i=1:rr1

% rectangle('Position',n1(i,:),'edgecolor','red');

% end

% for i=1:rr1

% RR=imcrop(R,n1(i,:));

% GG=imcrop(G,n1(i,:));

% BB=imcrop(B,n1(i,:));

% GLCM2 = graycomatrix(RR,'Offset',[2 0;0 2]);

% stats1 = graycoprops(GLCM2,{'contrast','homogeneity','Energy','Correlation'});

% ss1=cat(2,stats1.Contrast,stats1.Homogeneity,stats1.Energy,stats1.Correlation);

% GLCM2 = graycomatrix(GG,'Offset',[2 0;0 2]);

% stats2 = graycoprops(GLCM2,{'contrast','homogeneity','Energy','Correlation'});

% ss2=cat(2,stats2.Contrast,stats2.Homogeneity,stats2.Energy,stats2.Correlation);

GLCM2 = graycomatrix(bw1,'Offset',[2 0;0 2]);

stats3 = graycoprops(GLCM2,{'contrast','homogeneity','Energy','Correlation'});

TrainFM(index2,:)=cat(2,stats3.Contrast,stats3.Homogeneity,stats3.Energy,stats3.Correlation);

% end

end

% TESTING PART

mm='C:\Users\vysha\Desktop\malaria detection\Test\'; % TO READ TESTING SAMPLES

DatabasePath = mm;

TIFfile = dir(fullfile(DatabasePath, '\\*.jpg'));

for index2 = 1 : length(TIFfile)

fileName = fullfile(DatabasePath, TIFfile(index2).name);

I=imread(fileName);

R=I(:,:,1);

G=I(:,:,2);

B=I(:,:,3);

[m n]=size(R);

for i=1:m

for j=1:n

if R(i,j)>200

R(i,j)=0;

G(i,j)=0;

B(i,j)=0;

end

end

end

v=cat(3,R,G,B);

bw=edge(rgb2gray(v),'canny');

se = strel('disk',1);

bw2 = imdilate(bw,se);

bw1=imfill(bw2,'holes');

bw1=bwareaopen(bw1,100);

stats=regionprops(bw1,'all');

n1=cat(1,stats.BoundingBox);

% [rr1 rr2]=size(n1);

% figure, imshow(I)

% hold on

% for i=1:rr1

% rectangle('Position',n1(i,:),'edgecolor','red');

% end

% for i=1:rr1

% RR=imcrop(R,n1(i,:));

% GG=imcrop(G,n1(i,:));

% BB=imcrop(B,n1(i,:));

% GLCM2 = graycomatrix(RR,'Offset',[2 0;0 2]);

% stats1 = graycoprops(GLCM2,{'contrast','homogeneity','Energy','Correlation'});

% ss1=cat(2,stats1.Contrast,stats1.Homogeneity,stats1.Energy,stats1.Correlation);

% GLCM2 = graycomatrix(GG,'Offset',[2 0;0 2]);

% stats2 = graycoprops(GLCM2,{'contrast','homogeneity','Energy','Correlation'});

% ss2=cat(2,stats2.Contrast,stats2.Homogeneity,stats2.Energy,stats2.Correlation);

GLCM2 = graycomatrix(bw1,'Offset',[2 0;0 2]);

stats3 = graycoprops(GLCM2,{'contrast','homogeneity','Energy','Correlation'});

TestFM(index2,:)=cat(2,stats3.Contrast,stats3.Homogeneity,stats3.Energy,stats3.Correlation);

% end

end

save('TestFM','TrainFM','GT\_Train');

save('TestFM','TestFM','GT\_Test');

tic

Accuracy=classify\_malaria()

TotalTime=toc;

num1=sprintf('\n \n Accuracy = %f \n\n Time taken to classify %d samples is = %d',Accuracy,length(TestFM),TotalTime);

popbox(num1);

% --- Executes on button press in group execution of combined algorithm.

function pushbutton5\_Callback(hObject, eventdata, handles)

% hObject handle to pushbutton5 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

babesiosisDir=uigetdir('C:\Users\vysha\Desktop\malaria detection\');

imgSet = imageSet(babesiosisDir)

methods(imgSet)

editorwindow

%% Create a display of all Plasmodium images

togglefig('Plasmodium Images')

ax = gobjects(imgSet.Count,1);

for ii = 1:imgSet.Count

ax(ii) =...

subplot(floor(sqrt(imgSet.Count)),ceil(sqrt(imgSet.Count)),ii);

[~,currName] = fileparts(imgSet.ImageLocation{ii});

imshow(read(imgSet,ii))

title([num2str(ii),') ' currName],...

'interpreter','none','fontsize',7)

end

expandAxes(ax);

%%

targetImage = getimage(ax(2));

%%

togglefig('Plasmodium Images',true)

refreshImages

for ii = 1:imgSet.Count

mask = refinedMask(getimage(ax(ii)));

showMaskAsOverlay(0.5,mask,'b',[],ax(ii))

drawnow

end

expandAxes(ax);

%%

togglefig('Plasmodium Images')

for ii = 1:imgSet.Count

tmpMask = refinedMask2(imgSet,ii);

showMaskAsOverlay(0.5,tmpMask,'g',true,ax(ii))

drawnow

end

expandAxes(ax);

%%

detectCircles=@(x)imfindcircles(x,[20,35],'Sensitivity',0.89,'EdgeThreshold',0.04,'Method','TwoStage','ObjectPolarity','Dark');

%% Again, we check to see how robust the approach is:

togglefig('Plasmodium Images')

for ii = 1:imgSet.Count

[centers,radii] = detectCircles(rgb2gray(read(imgSet,ii)));

delete(findall(ax(ii),'type','line'))

viscircles(ax(ii),centers,radii,'edgecolor','b')

drawnow

end

expandAxes(ax);

%% We're almost there! Now we need only quantify infection...

infectionThreshold=135;

%% Is this more generalizable?

togglefig('Plasmodium Images',true)

refreshImages;

drawnow

%

for ii = 1:imgSet.Count

[pctInfection,centers,radii,isInfected,infectionMask] = ...

testForInfection(getimage(ax(ii)),targetImage,...

infectionThreshold,detectCircles);

title(ax(ii),...

['Pct Infection: ', num2str(pctInfection,2),...

' (' num2str(sum(isInfected)),...

' of ' num2str(numel(isInfected)) ')']);

viscircles(ax(ii),centers,radii,'edgecolor','b')

% createCirclesMask

infectedCellsMask = createCirclesMask(targetImage,...

centers(isInfected,:),...

radii(isInfected));

showMaskAsOverlay(0.3,infectedCellsMask,'g',ax(ii),false);

showMaskAsOverlay(0.5,infectionMask,'r',ax(ii),false);

drawnow

end

expandAxes(ax);

% --- Executes on button press in Accuracy of combined algorithm.

function pushbutton6\_Callback(hObject, eventdata, handles)

% hObject handle to pushbutton6 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

%ParasitologyDemo();

classificationLearner