```
function varargout = CTIMRX(varargin)
% CTIMRX MATLAB code for CTIMRX.fig
%      CTIMRX, by itself, creates a new CTIMRX or raises the existing
%      singleton*.
%
%      H = CTIMRX returns the handle to a new CTIMRX or the handle to
%      the existing singleton*.
%
%      CTIMRX('CALLBACK',hObject,eventData,handles,...) calls the
 local
%      function named CALLBACK in CTIMRX.M with the given input
 arguments.
%
%      CTIMRX('Property','Value',...) creates a new CTIMRX or raises
 the
%      existing singleton*.  Starting from the left, property value
 pairs are
%      applied to the CTIMRX before CTIMRX_OpeningFcn gets called.  An
%      unrecognized property name or invalid value makes property
 application
%      stop.  All inputs are passed to CTIMRX_OpeningFcn via varargin.
%
%      *See CTIMRX Options on GUIDE's Tools menu.  Choose "CTIMRX
 allows only one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

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

% Last Modified by GUIDE v2.5 10-May-2020 16:04:01

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @CTIMRX_OpeningFcn, ...
                   'gui_OutputFcn',  @CTIMRX_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 CTIMRX is made visible.
function CTIMRX_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 CTIMRX (see VARARGIN)

% Choose default command line output for CTIMRX
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);
axes(handles.axes4);
imshow('ISEC_logotipo.png');
% UIWAIT makes CTIMRX wait for user response (see UIRESUME)
% uiwait(handles.CTIMRX);

% --- Outputs from this function are returned to the command line.
function varargout = CTIMRX_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 pbImagemXRay.
function pbImagemXRay_Callback(hObject, eventdata, handles)
% hObject    handle to pbImagemXRay (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
cla(handles.axes2);
cla(handles.axes3);
cla(handles.axes7);
global filename;
global selectedImageG;
global pathname;
global GRAYimage;
global atv;
global selectedImage;
global newfilname;

%SELECIONAR A PASTA DE FOTOS
[filename, pathname] =
 uigetfile({'*.jpg';'*.png';'*.gif*';'*.tif'},'Pick an image ');

if filename
    %ENTRA NESTE IF SE UMA IMAGEM FOR SELECIONADA
    selectedImage = imread(strcat(pathname,'\', filename));

    %VERIFICAÇÃO SE A IMAGEM É RGB OU CINZA
```

```
    RGBorGRAY = ndims(selectedImage);  %devolve 2 ou 3 para saber se a
 imagem é a cor ou cinzento

    if RGBorGRAY == 2
        GRAYimage = selectedImage;
    else if RGBorGRAY == 3
            GRAYimage = rgb2gray(selectedImage);
        end
    end

    selectedImageG=imadjust(GRAYimage);

    axes(handles.axes1);
    imshow(selectedImageG);
    set(handles.edit5, 'enable', 'off');
    set(handles.edit5,'String',filename);
    atv=1;
    IMGtotal();
else
    %SE NENHUMA IMAGEM FOR SELECIONA APARECE UMA MESSAGE BOX
    msgbox('Nenhuma imagem selecionada, operação cancelada!');
end


% --- Executes on button press in pbSair.
function pbSair_Callback(hObject, eventdata, handles)
% hObject    handle to pbSair (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
SN=questdlg('Deseja sair?','SAIR','Sim','Não','Sim');
if strcmp(SN,'Não') %vai comparar duas strings
    return;
    %sai automaticamente e não passa ao delete, se for falso passa ao
 delete
end
delete(handles.CTIMRX);


% --- Executes during object deletion, before destroying properties.
function CTIMRX_DeleteFcn(hObject, eventdata, handles)
% hObject    handle to CTIMRX (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)


function CTIMRX_CloseRequestFcn(hObject, eventdata, handles)
% hObject    handle to projeto (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: delete(hObject) closes the figure
SN=questdlg('Deseja sair?','SAIR','Sim','Não','Sim');
if strcmp(SN,'Não') %vai comparar duas strings
    return;
```

```
    %sai automaticamente e não passa ao delete, se for falso passa ao
 delete
end
delete(handles.CTIMRX);


% --- Executes on button press in pbLimpar.
function pbLimpar_Callback(hObject, eventdata, handles)
% hObject    handle to pbLimpar (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
cla(handles.axes1);
cla(handles.axes2);
cla(handles.axes3);
cla(handles.axes7);

% --- Executes on button press in pbClassificar.
function pbClassificar_Callback(hObject, eventdata, handles)
% hObject    handle to pbClassificar (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global RGBimg;
global newfilename;
global filename;
global a;

extra(handles);
textura(handles);
axes(handles.axes3);
imshow(RGBimg);
newfilename=filename(1:end-11)

if a==1
    img = imread(strcat(newfilename,'2.png'));
    axes(handles.axes7);
    imshow(img);
else
    img = imread(strcat(newfilename,'1.png'));
    axes(handles.axes7);
    imshow(img);
end

msgbox('Finish');


% --- Executes on button press in rbCinzento8.
function rbCinzento8_Callback(hObject, eventdata, handles)
% hObject    handle to rbCinzento8 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of rbCinzento8
```

```
% --- Executes on button press in rbCinzento16.
function rbCinzento16_Callback(hObject, eventdata, handles)
% hObject    handle to rbCinzento16 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of rbCinzento16

% --- Executes on button press in rbClasse4.
function rbClasse2_Callback(hObject, eventdata, handles)
% hObject    handle to rbClasse4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of rbClasse4

function eValor_Callback(hObject, eventdata, handles)
% hObject    handle to eValor (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 eValor as text
%        str2double(get(hObject,'String')) returns contents of eValor
 as a double

% --- Executes during object creation, after setting all properties.
function eValor_CreateFcn(hObject, eventdata, handles)
% hObject    handle to eValor (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

function extra(handles)
global ZonaCortada;
global nCinzento;
global offset;
global C;

%CLASSES
valorC=get (handles.uiClasses,'SelectedObject');
escolhaC=find([handles.rbClasse2,...
               handles.rbClasse3]==valorC);
classe2=1;
classe3=2;

switch escolhaC
    case classe2
```

```
        C=2;
    case classe3
        C=3;
end

%NIVEIS DE CINZENTO
valorN=get (handles.uiCinzento,'SelectedObject');
escolhaN=find([handles.rbCinzento8,...
               handles.rbCinzento16]==valorN);
NC8=1;
NC16=2;

switch escolhaN
    case NC8
        nCinzento=8;
    case NC16
        nCinzento=16;
end

%ANGULO
valorA=get (handles.uiAngulo,'SelectedObject');
escolhaA=find([handles.rbGraus0,...
               handles.rbGraus45,...
               handles.rbGraus90,...
               handles.rbGraus135]==valorA);
graus0=1;
graus45=2;
graus90=3;
graus135=4;
switch escolhaA
    case graus0
        offset=[0 1];
    case graus45
        offset=[-1 1];
    case graus90
        offset=[-1 0];
    case graus135
        offset=[-1 -1];
end

% --------------------------------------------------------------------
function menu_Ficheiros_Callback(hObject, eventdata, handles)
% hObject    handle to menu_Ficheiros (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% --------------------------------------------------------------------
function menu_ajuda_Callback(hObject, eventdata, handles)
% hObject    handle to menu_ajuda (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% --------------------------------------------------------------------
function menu_GuiaUtilizador_Callback(hObject, eventdata, handles)
```

```
% hObject    handle to menu_GuiaUtilizador (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
GuiaUtilizador();

% --------------------------------------------------------------------
function menu_relatorio_Callback(hObject, eventdata, handles)
% hObject    handle to menu_relatorio (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
open('CARACTERÍSTICAS-DE-TEXTURA-EM-IMAGENS-MÉDICAS-DE-RAIO-X.pdf');

% --- Executes on button press in pbZonaSelecionada.
function pbZonaSelecionada_Callback(hObject, eventdata, handles)
% hObject    handle to pbZonaSelecionada (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global selectedImageG;
cla(handles.axes2);
cla(handles.axes3);
cla(handles.axes7);
clear ZonaCortada;
axes(handles.axes1);
imshow(selectedImageG);

global ZonaCortada;

%FUNÇÃO PARA SELECIONAR UMA CERTA REGIÃO DA IMAGEM
[mask, xi, yi] = roipoly(selectedImageG);
tamanho=size(xi);
if tamanho(1)<6
    x=xi(1); %coordenadas primeiro vértice
    y=yi(1);
    width = abs(xi(2)-x); %distância x entre a coordenada x do
 terceiro e primeiro vértices
    height = abs(yi(2)-yi(4)); %distância y entre a coordenada x do  e
 primeiro vértices
    rect=[x, y, width, height];
    ZonaCortada = imcrop(selectedImageG,rect);
    axes(handles.axes2);
    imshow(ZonaCortada);
else
    %SE NENHUMA ZONA DA IMAGEM FOR SELECIONA APARECE UMA MESSAGE BOX
    msgbox('Nenhuma zona selecionada, operação cancelada!');
end

function textura(handles)
global ZonaCortada;
global nCinzento;
global offset;
global C;
global clas;
global RGBimg;
global filename;
```

```
global a;

nExtra=1;
[lin col]=size(ZonaCortada);

%cria uma nova imagem a partir da imagem da zona selecionada e
 acrescenta
%uma linha em cima e embaixo e uma coluna ao inicio e outra ao fim
newZona=padarray(ZonaCortada,[nExtra nExtra],'symmetric','both');

%CALCULAR TEXTURAS DE HARALICK
for i=1:col
    for j=1:lin
         Iroi=newZona(j:j+nExtra,i:i+nExtra); %
         if isempty(find(isnan(Iroi)==1, 1)) %só entra se Iroi não
 tiver valores NAN
             glcm = graycomatrix(Iroi, 'Offset',
 offset, 'NumLevels',nCinzento,'Symmetric', true);
             % calcula as texturas
             sprintf('%d i, %d j',i,j)
             texturas((i-1)*lin+j,:) = haralickTextureFeatures(glcm)';
         end
    end
end

save('output','texturas');

valor=get (handles.uiClassificador,'SelectedObject');
escolha=find([handles.rbLinear,...
              handles.rbSVM,...
              handles.rbEnsemble]==valor);

%VAI BUSCRA A RESPETIVA TABELA PARA COMPARAR
if nCinzento==8
    niveis='8';
else
    niveis='16';
end

if offset(1)==0 && offset(2)==1
    graus='0';
else if offset(1)==-1 && offset(2)==1
        graus='45';
    else if offset(1)==-1 && offset(2)==0
            graus='90';
        else if offset(1)==-1 && offset(2)==-1
            graus='135';
            end
        end
    end
end

s=dir(['tabela_',graus,'graus_',niveis,'niveis.xlsx']);
tabela=importdata(s.name);
```

```
if C==2
    classe='2';
    X=tabela.data((2:14),[4,6,7]);
    Y=tabela.textdata((2:14),1);
    %imagem a identificar as cores
else if C==3
        classe='3';
        X=tabela.data(:,[4,6,7]);  %vai buscar à tabela os valores na
 coluna 4,6 e 7
        Y=tabela.textdata((2:16),1);   %vai buscar o que está na
 coluna 1 de todas as linhas
    end
end
disp('antes classificador');
switch escolha
    case 1
        clas='1';
        classificador=fitcdiscr(X,Y,'DiscrimType','pseudolinear');
    case 2
        clas='2';
        classificador=fitcecoc(X,Y);
    case 3
        clas='3';

 classificador=fitensemble(X,Y,'Bag',100,'Tree','Type','Classification');
end


%CLASSIFICA E VÊ QUAL É A CLASSE
out=load('output.mat');
T=out.texturas(:,[4,6,7]);
resultado=predict(classificador,T);

%RECONSTRÓI A ZONA CORTADA COM O TAMANHO ORIGINAL
finalIMG=reshape(resultado,[lin,col]);

%IMAGEM A CORES
RGBimg=zeros(lin,col,3);
a=0;
disp('antes classificacao');
%IMAGEM FINAL A CORES
for i=1:col
    for j=1:lin
            if strcmp(finalIMG(j,i),'C1')
                RGBimg(j,i,:)=ind2rgb(ZonaCortada(j,i),[0,0,1]); %osso
 fica a azul
            else if strcmp(finalIMG(j,i),'C2')
                RGBimg(j,i,:)=ind2rgb(ZonaCortada(j,i),
[0,1,1]); %tecido fica a azul mais claro
                 else
                    RGBimg(j,i,:)=ind2rgb(ZonaCortada(j,i),
[1,0,0]); %fundo fica a vermelho
                    a=1;
```

```
                end
            end
    end
end

% --- Executes on button press in rbClasse3.
function rbClasse3_Callback(hObject, eventdata, handles)
% hObject    handle to rbClasse3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of rbClasse3


function edit5_Callback(hObject, eventdata, handles)
% hObject    handle to edit5 (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 edit5 as text
%        str2double(get(hObject,'String')) returns contents of edit5
 as a double


% --- Executes during object creation, after setting all properties.
function edit5_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit5 (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 pbClassificaTudo.
function pbClassificaTudo_Callback(hObject, eventdata, handles)
% hObject    handle to pbClassificaTudo (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
cla(handles.axes2);
cla(handles.axes3);
cla(handles.axes7);
global ZonaCortada;
global selectedImageG;
global atv;

SN=questdlg('A interface vai demorar, deseja
 continuar?','CONTINUAR','Sim','Não','Sim');
if strcmp(SN,'Não') %vai comparar duas strings
```

```
    %volta para a GUI
else
    B=selectedImageG(1:2:end,1:2:end,:);
     C=B(1:2:end,1:2:end,:);
    ZonaCortada=imadjust(C);
    extra(handles);
    textura(handles);
    atv=2;
    IMGtotal();
end
msgbox('Acabou');

% --------------------------------------------------------------------
function menuSobre_Callback(hObject, eventdata, handles)
% hObject    handle to menuSobre (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)


% --------------------------------------------------------------------
function menu_autores_Callback(hObject, eventdata, handles)
% hObject    handle to menu_autores (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
Autores();

% --------------------------------------------------------------------
function menu_versao_Callback(hObject, eventdata, handles)
% hObject    handle to menu_versao (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
open('SobreMatlab.txt');

% --------------------------------------------------------------------
function menu_detalhado_Callback(hObject, eventdata, handles)
% hObject    handle to menu_detalhado (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
open('GuiaUtilizador.pdf');

% --- Executes on button press in pbGuardarIMG.
function pbGuardarIMG_Callback(hObject, eventdata, handles)
% hObject    handle to pbGuardarIMG (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global filename;
global classe;
global niveis;
global graus;
global clas;
global pathname;

frame = getframe(handles.axes3);
novaIMG = frame2im(frame);
```

```
%GUARDAR IMAGEM CARACTERIZADA
name=filename(1:end-4);
NEWfilename=['img',name,'_nclass',classe,'_niveis',niveis,'_graus',graus,'_clas',c
Path = fullfile(pathname,'Imagens guardadas\',NEWfilename);
imwrite(novaIMG, Path);
```

Published with MATLAB® R2016a