GUI

function varargout = Gui1(varargin)

%GUI1 M-file for Gui1.fig

% GUI1, by itself, creates a new GUI1 or raises the existing

% singleton\*.

%

% H = GUI1 returns the handle to a new GUI1 or the handle to

% the existing singleton\*.

%

% GUI1('Property','Value',...) creates a new GUI1 using the

% given property value pairs. Unrecognized properties are passed via

% varargin to Gui1\_OpeningFcn. This calling syntax produces a

% warning when there is an existing singleton\*.

%

% GUI1('CALLBACK') and GUI1('CALLBACK',hObject,...) call the

% local function named CALLBACK in GUI1.M with the given input

% arguments.

%

% \*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 Gui1

% Last Modified by GUIDE v2.5 07-Apr-2016 14:47:27

% Begin initialization code - DO NOT EDIT

gui\_Singleton = 1;

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

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

'gui\_OpeningFcn', @Gui1\_OpeningFcn, ...

'gui\_OutputFcn', @Gui1\_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 Gui1 is made visible.

function Gui1\_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 unrecognized PropertyName/PropertyValue pairs from the

% command line (see VARARGIN)

% Choose default command line output for Gui1

handles.output = hObject;

% Update handles structure

guidata(hObject, handles);

% UIWAIT makes Gui1 wait for user response (see UIRESUME)

% uiwait(handles.figure1);

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

function varargout = Gui1\_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 pushbutton5.

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)

Vmon();

% global vidIN;

%

% axes(handles.axes1);

% vid = image(zeros(480,640,3),'parent',handles.axes1);

% step(vidIN,vid)

% --- Executes on button press in pushbutton6.

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)

global down

down=1;

% --- Executes on button press in pushbutton7.

function pushbutton7\_Callback(hObject, eventdata, handles)

% hObject handle to pushbutton7 (see GCBO)

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

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

global up

up=1;

% --- Executes on button press in pushbutton1.

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)

global a;

a= videoinput('winvideo',1,'YUY2\_640x480');

axes(handles.axes1);

vid = image(zeros(480,640,3),'parent',handles.axes1);

preview(a,vid)

% --- Executes on button press in pushbutton2.

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)

global a;

global c;

c=ycbcr2rgb(getsnapshot(a));

axes(handles.axes1);

imshow(c);

% --- Executes on button press in pushbutton3.

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)

global c;

datapath='C:\Users\Dell-pc\Vm\New folder';

recognized\_img = facerecog( c,datapath );

axes(handles.axes2);

imshow(recognized\_img)

im='match found';

set(handles.text2,'String',im);

% --- Executes on button press in checkbox5.

function checkbox5\_Callback(hObject, eventdata, handles)

% hObject handle to checkbox5 (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 checkbox5

global v

v = get(hObject,'Value');

% --- Executes on button press in checkbox6.

function checkbox6\_Callback(hObject, eventdata, handles)

% hObject handle to checkbox6 (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 checkbox6

global rm

rm = get(hObject,'Value');

% --- Executes on button press in checkbox9.

function checkbox9\_Callback(hObject, eventdata, handles)

% hObject handle to checkbox9 (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 checkbox9

global tu

tu = get(hObject,'Value');

% --- Executes on button press in checkbox11.

function checkbox11\_Callback(hObject, eventdata, handles)

% hObject handle to checkbox11 (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 checkbox11

global bm

bm = get(hObject,'Value');

% --- Executes on button press in checkbox12.

function checkbox12\_Callback(hObject, eventdata, handles)

% hObject handle to checkbox12 (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 checkbox12

global gm

gm = get(hObject,'Value');

function [ recognized\_img ] = facerecog( test\_img,datapath )

%UNTITLED Summary of this function goes here

% Detailed explanation goes here

D=dir(datapath);

imgcount=0;

for i=1: size(D,1)

if not(strcmp(D(i).name,'.')|strcmp(D(i).name,'..')|strcmp(D(i).name,'Thumbs.db'))

imgcount=imgcount+1;

end

end

X=[];

for i=1 : imgcount

str=strcat(datapath,'\',int2str(i),'.jpg');

img=imread(str);

img=rgb2gray(img);

[r c]=size(img);

temp=reshape(img',r\*c,1);

X=[X temp];

end

m=mean(X,2);

imgcount=size(X,2);

A=[];

for i=1 : imgcount

temp=double(X(:,i))-m;

A=[A temp];

end

L=A'\*A;

[V,D]=eig(L);

L\_eig\_vec=[];

for i=1 : size(V,2)

if (D(i,i)>1)

L\_eig\_vec=[L\_eig\_vec V(:,i)];

end

end

eigenfaces=A\*L\_eig\_vec;

projectimg=[];

for i=1:size(eigenfaces,2)

temp=eigenfaces'\*A(:,i);

projectimg=[projectimg temp];

end

%Testing

test\_image=test\_img (: ,: ,1);

[r c]=size(test\_image);

temp=reshape(test\_image',r\*c,1);

temp=double(temp)-m;

projtestimg=eigenfaces'\*temp;

global euclide\_dist;

euclide\_dist=[];

for i=1:size(eigenfaces,2)

temp=(norm(projtestimg-projectimg(:,i)))^2;

euclide\_dist=[euclide\_dist temp];

end

[ euclide\_dist\_min recognized\_index]=min( euclide\_dist);

recognized\_img=strcat(datapath,'\',int2str(recognized\_index),'.jpg');

end

function [ ] = Vmon()

%UNTITLED7 Summary of this function goes here

% Detailed explanation goes here

imaqreset;

videoPlayer=vision.VideoPlayer('Name','mouse','Position',[100 100 640 480]);

a=imaq.VideoDevice('winvideo',1,'YUY2\_640x480','ReturnedColorSpace','RGB');

global rth

global gth

global bth

rth=0.15;

gth=0.09;

bth=0.18;

hblobrb = vision.BlobAnalysis('AreaOutputPort',false,'CentroidOutputPort',true,'BoundingBoxOutputPort',true,'MaximumBlobArea',2000,'MinimumBlobArea',300,'MaximumCount',1);

hshapeinsBox=vision.ShapeInserter('BorderColorSource','Input port',...

'Fill',true,...

'FillColorSource','Input port',...

'Opacity',0.4);

hblobg= vision.BlobAnalysis('AreaOutputPort',false,'CentroidOutputPort',true,'BoundingBoxOutputPort',true,'MaximumBlobArea',2000,'MinimumBlobArea',300,'MaximumCount',3);

gcount=0;

gdcount=0;

grcount=0;

bcount=0;

bdcount=0;

brcount=0;

global v;

% global g1;

% global b1;

sure=5;

M= java.awt.Robot;

if v==1

global rm;

global bm;

global gm;

global tu;

global up;

global down;

while(v)

if tu==1

if rm==1

if up==1

rth = rth+0.01;

up=0;

end

end

if bm==1

if up==1

bth = bth+0.01;

up=0;

end

end

if gm==1

if up==1

gth = gth+0.01;

up=0;

end

end

end

if tu==1

if rm==1

if down==1

rth = rth-0.01;

down=0;

end

end

if bm==1

if down==1

bth = bth-0.01;

down=0;

end

end

if gm==1

if down==1

gth = gth-0.01;

down=0;

end

end

end

c=step(a);

w=flipdim(c,2);

r= w(:,:,1);

gr= w(:,:,2);

b= w(:,:,3);

g= rgb2gray(w);

red=imsubtract(r,g);

rthf=im2bw(red,rth);

r1= imfill(rthf,'holes');

r2= medfilt2(r1,[5,5]);

[Cr,rbox]=step(hblobrb,r2);

green=imsubtract(gr,g);

gthf=im2bw(green,gth);

g1= imfill(gthf,'holes');

g2= medfilt2(g1,[5,5]);

[Cg,gbox]=step(hblobg,g2);

blue=imsubtract(b,g);

bthf=im2bw(blue,bth);

b1= imfill(bthf,'holes');

b2= medfilt2(b1,[5,5]);

[Cb,bbox]=step(hblobg,b2);

if tu==0

if rm==1

if length(rbox(:,1))==1

x=Cr(1)\*2.16;

y=Cr(2)\*1.8-15;

M.mouseMove(x,y);

end

end

end

if tu==0

if bm==1

if ~isempty(bbox)

if (length(bbox(:,1))==1)

gcount=gcount+1;

if gcount == sure

M.mousePress(16);

pause(0.1);

M.mouseRelease(16);

gcount=0;

end

end

end

% if (length(gbox(:,1))==2)

% gdcount=gdcount+1;

% if gdcount == sure

%

% M.mousePress(16);

% pause(0.1);

% M.mouseRelease(16);

% pause(0.2)

% M.mousePress(16);

% pause(0.1);

% M.mouseRelease(16);

% gdcount=0;

% end

% end

end

end

if tu==0

if gm==1

if length((gbox(:,1))==1)

gdcount=0;

grcount=grcount+1;

if grcount == sure

M.mousePress(4);

pause(0.1);

M.mouseRelease(4);

grcount=0;

end

end

% if length(gbox(:,1))==1

%

%

% M.mousePress(16);

%

% else

% M.mouseRelease(16);

%

%

% end

end

end

global vidIN;

vidIN=step(hshapeinsBox, w,rbox,single([1 0 0]));

vidIN=step(hshapeinsBox, vidIN,gbox,single([0 1 0]));

vidIN=step(hshapeinsBox, vidIN,bbox,single([0 0 1]));

step(videoPlayer,vidIN);

end

end

end