**GUI.M**

function varargout = gui(varargin)

% GUI MATLAB code for gui.fig

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

% singleton\*.

%

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

% the existing singleton\*.

%

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

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

%

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

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

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

% unrecognized property name or invalid value makes property application

% stop. All inputs are passed to gui\_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 gui

% Last Modified by GUIDE v2.5 28-Apr-2022 12:51:29

% Begin initialization code - DO NOT EDIT

gui\_Singleton = 1;

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

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

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

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

function gui\_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 gui (see VARARGIN)

% Choose default command line output for gui

handles.output = hObject;

a=zeros([256 256]);

axes(handles.axes1);

imshow(a);

axes(handles.axes2);

imshow(a);

axes(handles.axes3);

imshow(a);

% Update handles structure

guidata(hObject, handles);

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

% uiwait(handles.figure1);

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

function varargout = gui\_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;

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)

global train

matlabpath ='C:\Users\karth\Desktop\tuber matlab\'

data = fullfile(matlabpath,'dataset')

train = imageDatastore(data, 'IncludeSubfolders',true,'LabelSource','foldernames');

count = train.countEachLabel;

msgbox('Dataset Loaded Successfully')

% Update handles structure

guidata(hObject, handles);

% --- 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 layers

disp('Pre-Trained Model Loaded...')

net = googlenet;

deepNetworkDesigner(net)

layers = [ imageInputLayer([400 400 3])

net(2:end-3)

fullyConnectedLayer(2)

softmaxLayer

classificationLayer()

]

msgbox('Pre-Trained Model Loaded Successfully')

% Update handles structure

guidata(hObject, handles);

% --- 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)

%training

global opt training train layers

opt = trainingOptions('adam', ...

'InitialLearnRate', 0.001, ...

'LearnRateSchedule', 'piecewise', ...

'LearnRateDropFactor', 0.1, ...

'LearnRateDropPeriod', 8, ...

'L2Regularization', 0.004, ...

'MaxEpochs', 10, ...

'MiniBatchSize', 100, ...

'Verbose', true, 'Plots','training-progress');

training = trainNetwork(train,layers,opt);

msgbox('Trained Completed')

% Update handles structure

guidata(hObject, handles);

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 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 inp

cd input

[file path] = uigetfile('\*.bmp;\*.jpg;\*.jpeg','Pick an Image File');

if isequal(file,0)

warndlg('File not selected');

else

inp = imread(file);

cd ..

axes(handles.axes1);

imshow(inp);

img=inp;

if size(inp,3)>1

Freg =im2gray(inp);

end

handles.img=img;

end

% Update handles structure

guidata(hObject, handles);

% --- 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 inp J

img = im2gray(inp);

J = medfilt2(img);

axes(handles.axes2);

title('Filtered Image');

imshow(J);

% Update handles structure

guidata(hObject, handles);

% --- Executes on button press in pushbutton4.

% --- 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)

global training inp out

out = classify(training,inp);

axes(handles.axes3);

imshow(inp);

title(string(out));

% Update handles structure

guidata(hObject, handles);

if (string(out)=="Normal")

msgbox(string(out));

else

msgbox(string(out));

[segout] = seg(inp);

boundary = bwboundaries(im2bw(segout));

end

% out = classify(training,inp);

% yval = inp;

% accuracy = mean()

**SEG.M**

function [inp,J] = seg(inp, J)

%SEG Summary of this function goes here

% Detailed explanation goes here

% This is a program for extracting objects from an image. Written for vehicle number plate segmentation and extraction

% Authors : Jeny Rajan, Chandrashekar P S

% U can use attached test image for testing

% input - give the image file name as input. eg :- car3.jpg

%k=input('Enter the file name','s'); % input image; color image

%im=imread(inp);

im1=rgb2gray(inp);

im1=medfilt2(im1,[3 3]); %Median filtering the image to remove noise%

BW = edge(im1,'sobel'); %finding edges

[imx,imy]=size(BW);

msk=[0 0 0 0 0;

0 1 1 1 0;

0 1 1 1 0;

0 1 1 1 0;

0 0 0 0 0;];

B=conv2(double(BW),double(msk)); %Smoothing image to reduce the number of connected components

L = bwlabel(B,8);% Calculating connected components

mx=max(max(L))

% There will be mx connected components.Here U can give a value between 1 and mx for L or in a loop you can extract all connected components

% If you are using the attached car image, by giving 17,18,19,22,27,28 to L you can extract the number plate completely.

[r,c] = find(L==17);

rc = [r c];

[sx sy]=size(rc);

n1=zeros(imx,imy);

for i=1:sx

x1=rc(i,1);

y1=rc(i,2);

n1(x1,y1)=255;

end % Storing the extracted image in an array

figure,imshow(inp);

figure,imshow(B);

%figure,imshow(n1,[]);

end