classdef A\_Novel\_Method\_of\_classification\_of\_Tumor\_Using\_CNN\_exported < matlab.apps.AppBase

% Properties that correspond to app components

properties (Access = public)

ANovelMethodofClassificationofTumorusingCNNUIFigure matlab.ui.Figure

usermanual matlab.ui.container.Panel

Label\_5 matlab.ui.control.Label

Label\_4 matlab.ui.control.Label

Label\_3 matlab.ui.control.Label

Label\_2 matlab.ui.control.Label

Label matlab.ui.control.Label

HomeButton matlab.ui.control.StateButton

TumorSpotButton matlab.ui.control.StateButton

TumorTypeButton matlab.ui.control.StateButton

BenignorMalignantButton matlab.ui.control.StateButton

ResultPanel matlab.ui.container.Panel

Output\_box matlab.ui.control.EditField

ThegivenMRIscanconsistsofLabel matlab.ui.control.Label

MRIscanimagegivenLabel matlab.ui.control.Label

Input\_img matlab.ui.control.Image

ImageButton matlab.ui.control.Button

ClickontheimagebuttonandselecttheMRIimageoftheBrainLabel matlab.ui.control.Label

tumorspot\_axes matlab.ui.control.UIAxes

TumorDetectionandClassificationLabel matlab.ui.control.Label

end

% Callbacks that handle component events

methods (Access = private)

% Value changed function: BenignorMalignantButton

function BenignorMalignantButtonValueChanged(app, event)

value = app.BenignorMalignantButton.Value;

cla(app.tumorspot\_axes);

app.usermanual.Visible='off';

white\_screen=ones([256 256 3]);

app.Input\_img.ImageSource=white\_screen;

app.tumorspot\_axes.Visible='off';

app.Output\_box.Value=' ';

if strcmp(app.ResultPanel.Visible,'off')==1 && strcmp(app.ResultPanel.Enable,'off')==1

app.ResultPanel.Visible='on';

app.ResultPanel.Enable='on';

end

app.ResultPanel.Title='Benign Or Malignant';

end

% Value changed function: TumorTypeButton

function TumorTypeButtonValueChanged(app, event)

value = app.TumorTypeButton.Value;

cla(app.tumorspot\_axes);

app.usermanual.Visible='off';

white\_screen=ones([256 256 3]);

app.Input\_img.ImageSource=white\_screen;

app.Output\_box.Value=' ';

app.tumorspot\_axes.Visible='off';

app.Output\_box.Value=' ';

if strcmp(app.ResultPanel.Visible,'off')==1 && strcmp(app.ResultPanel.Enable,'off')==1

app.ResultPanel.Visible='on';

app.ResultPanel.Enable='on';

end

app.ResultPanel.Title='Tumor Type';

end

% Value changed function: TumorSpotButton

function TumorSpotButtonValueChanged(app, event)

value = app.TumorSpotButton.Value;

white\_screen=ones([256 256 3]);

app.Input\_img.ImageSource=white\_screen;

cla(app.tumorspot\_axes);

app.usermanual.Visible='off';

app.Output\_box.Visible='off';

app.Output\_box.Enable='off';

if strcmp(app.ResultPanel.Visible,'off')==1 && strcmp(app.ResultPanel.Enable,'off')==1

app.ResultPanel.Visible='on';

app.ResultPanel.Enable='on';

end

app.ResultPanel.Title='Tumor Spot';

end

% Button pushed function: ImageButton

function ImageButtonPushed(app, event)

disp(app.ResultPanel.Title);

app.Output\_box.Value=' ';

white\_screen=ones([256 256 3]);

app.Input\_img.ImageSource=white\_screen;

[file,path]=uigetfile('\*.\*',"MRI input image");

if isequal(file,0)

else

MRI\_scan=imread(fullfile(path,file));

app.Input\_img.ImageSource=MRI\_scan;

if strcmp(app.ResultPanel.Title,'Benign Or Malignant')==1

app.Output\_box.Visible='on';

app.Output\_box.Enable='on';

app.Output\_box.Value='Processing.........';

img2 = im2bw(MRI\_scan);

signal1 = img2(:,:);

[cA1,cH1,cV1,cD1] = dwt2(signal1,'db4');

[cA2,cH2,cV2,cD2] = dwt2(cA1,'db4');

[cA3,cH3,cV3,cD3] = dwt2(cA2,'db4');

DWT\_feat = [cA3,cH3,cV3,cD3];

G = pca(DWT\_feat);

g = graycomatrix(G);

stats = graycoprops(g,'Contrast Correlation Energy Homogeneity');

Contrast = stats.Contrast;

Correlation = stats.Correlation;

Energy = stats.Energy;

Homogeneity = stats.Homogeneity;

Mean = mean2(G);

Standard\_Deviation = std2(G);

Entropy = entropy(G);

RMS = mean2(rms(G));

%Skewness = skewness(img)

Variance = mean2(var(double(G)));

a = sum(double(G(:)));

Smoothness = 1-(1/(1+a));

Kurtosis = kurtosis(double(G(:)));

Skewness = skewness(double(G(:)));

% Inverse Difference Movement

m = size(G,1);

n = size(G,2);

in\_diff = 0;

for i = 1:m

for j = 1:n

temp = G(i,j)./(1+(i-j).^2);

in\_diff = in\_diff+temp;

end

end

IDM = double(in\_diff);

feat = [Contrast,Correlation,Energy,Homogeneity, Mean, Standard\_Deviation, Entropy, RMS, Variance, Smoothness, Kurtosis, Skewness, IDM];

load 'Trained data'\Trainset.mat

xdata = meas;

group = label;

svmStruct1 = fitcsvm(xdata,group,'KernelFunction', 'linear');

species= predict(svmStruct1,feat);

disp(species);

if strcmpi(species,'MALIGNANT')

app.Output\_box.Value='Malignant Tumor';

else

app.Output\_box.Value='Benign Tumor';

end

elseif strcmp(app.ResultPanel.Title,'Tumor Type')==1

pause(1);

app.Output\_box.Visible='on';

app.Output\_box.Enable='on';

app.Output\_box.Value='Processing.........';

load 'Trained data'\cnn87.mat

MRI\_scan=imresize(MRI\_scan,[224,224]);

[label,~]=classify(cnn87,MRI\_scan);

pause(1);

label=string(label);

if strcmp(label,'glioma\_tumor')==1

disp(label);

label='Glioma Tumor';

app.Output\_box.Value=label;

elseif strcmp(label,'meningioma\_tumor')==1

disp(label);

label=' Meningioma Tumor';

app.Output\_box.Value=label;

elseif strcmp(label,'no\_tumor')==1

disp(label);

label='No Tumor';

app.Output\_box.Value=label;

elseif strcmp(label,'pituitary\_tumor')==1

disp(label);

label='Pituitary Tumor'

app.Output\_box.Value=label

end

disp('ended');

elseif strcmp(app.ResultPanel.Title,'Tumor Spot')==1

cla(app.tumorspot\_axes);

pause(1);

I=MRI\_scan;

gray = rgb2gray(I);

img = im2bw(gray,.6);

img = bwareaopen(img,80);

TUMOr\_spot\_axes\_display=app.tumorspot\_axes;

imshow(img,'Parent',TUMOr\_spot\_axes\_display);

end

end

end

% Value changed function: HomeButton

function HomeButtonValueChanged(app, event)

value = app.HomeButton.Value;

app.usermanual.Visible='on';

app.tumorspot\_axes.Visible='off';

cla(app.tumorspot\_axes,"reset");

app.Output\_box.Visible='off';

end

end

% Component initialization

methods (Access = private)

% Create UIFigure and components

function createComponents(app)

% Create ANovelMethodofClassificationofTumorusingCNNUIFigure and hide until all components are created

app.ANovelMethodofClassificationofTumorusingCNNUIFigure = uifigure('Visible', 'off');

app.ANovelMethodofClassificationofTumorusingCNNUIFigure.Color = [0.9412 0.9412 0.9412];

app.ANovelMethodofClassificationofTumorusingCNNUIFigure.Position = [100 100 912 568];

app.ANovelMethodofClassificationofTumorusingCNNUIFigure.Name = 'A Novel Method of Classification of Tumor using CNN';

app.ANovelMethodofClassificationofTumorusingCNNUIFigure.Resize = 'off';

% Create TumorDetectionandClassificationLabel

app.TumorDetectionandClassificationLabel = uilabel(app.ANovelMethodofClassificationofTumorusingCNNUIFigure);

app.TumorDetectionandClassificationLabel.HorizontalAlignment = 'center';

app.TumorDetectionandClassificationLabel.FontSize = 17;

app.TumorDetectionandClassificationLabel.FontWeight = 'bold';

app.TumorDetectionandClassificationLabel.FontColor = [0.7176 0.2745 1];

app.TumorDetectionandClassificationLabel.Position = [1 527 912 41];

app.TumorDetectionandClassificationLabel.Text = 'Tumor Detection and Classification';

% Create ResultPanel

app.ResultPanel = uipanel(app.ANovelMethodofClassificationofTumorusingCNNUIFigure);

app.ResultPanel.Enable = 'off';

app.ResultPanel.ForegroundColor = [0 0 1];

app.ResultPanel.Visible = 'off';

app.ResultPanel.FontWeight = 'bold';

app.ResultPanel.Position = [72 53 770 374];

% Create tumorspot\_axes

app.tumorspot\_axes = uiaxes(app.ResultPanel);

app.tumorspot\_axes.Toolbar.Visible = 'off';

app.tumorspot\_axes.XColor = 'none';

app.tumorspot\_axes.XTick = [];

app.tumorspot\_axes.YColor = 'none';

app.tumorspot\_axes.YTick = [];

app.tumorspot\_axes.ZColor = 'none';

app.tumorspot\_axes.Visible = 'off';

app.tumorspot\_axes.Position = [436 66 208 203];

% Create ClickontheimagebuttonandselecttheMRIimageoftheBrainLabel

app.ClickontheimagebuttonandselecttheMRIimageoftheBrainLabel = uilabel(app.ResultPanel);

app.ClickontheimagebuttonandselecttheMRIimageoftheBrainLabel.Position = [58 321 350 22];

app.ClickontheimagebuttonandselecttheMRIimageoftheBrainLabel.Text = 'Click on the image button and select the MRI image of the Brain:';

% Create ImageButton

app.ImageButton = uibutton(app.ResultPanel, 'push');

app.ImageButton.ButtonPushedFcn = createCallbackFcn(app, @ImageButtonPushed, true);

app.ImageButton.BackgroundColor = [0.149 0.149 0.149];

app.ImageButton.FontColor = [0.9412 0.9412 0.9412];

app.ImageButton.Position = [431 316 92 33];

app.ImageButton.Text = 'Image';

% Create Input\_img

app.Input\_img = uiimage(app.ResultPanel);

app.Input\_img.Position = [119 73 200 188];

% Create MRIscanimagegivenLabel

app.MRIscanimagegivenLabel = uilabel(app.ResultPanel);

app.MRIscanimagegivenLabel.HorizontalAlignment = 'center';

app.MRIscanimagegivenLabel.FontColor = [1 0 0];

app.MRIscanimagegivenLabel.Position = [128 272 181 22];

app.MRIscanimagegivenLabel.Text = 'MRI scan image given';

% Create ThegivenMRIscanconsistsofLabel

app.ThegivenMRIscanconsistsofLabel = uilabel(app.ResultPanel);

app.ThegivenMRIscanconsistsofLabel.HorizontalAlignment = 'right';

app.ThegivenMRIscanconsistsofLabel.FontSize = 14;

app.ThegivenMRIscanconsistsofLabel.FontWeight = 'bold';

app.ThegivenMRIscanconsistsofLabel.Enable = 'off';

app.ThegivenMRIscanconsistsofLabel.Visible = 'off';

app.ThegivenMRIscanconsistsofLabel.Position = [426 186 225 22];

app.ThegivenMRIscanconsistsofLabel.Text = 'The given MRI scan consists of: ';

% Create Output\_box

app.Output\_box = uieditfield(app.ResultPanel, 'text');

app.Output\_box.Editable = 'off';

app.Output\_box.HorizontalAlignment = 'center';

app.Output\_box.Enable = 'off';

app.Output\_box.Visible = 'off';

app.Output\_box.Position = [436 126 215 51];

% Create BenignorMalignantButton

app.BenignorMalignantButton = uibutton(app.ANovelMethodofClassificationofTumorusingCNNUIFigure, 'state');

app.BenignorMalignantButton.ValueChangedFcn = createCallbackFcn(app, @BenignorMalignantButtonValueChanged, true);

app.BenignorMalignantButton.Text = 'Benign or Malignant';

app.BenignorMalignantButton.BackgroundColor = [0 0 0];

app.BenignorMalignantButton.FontColor = [0.9412 0.9412 0.9412];

app.BenignorMalignantButton.Position = [244 468 156 34];

% Create TumorTypeButton

app.TumorTypeButton = uibutton(app.ANovelMethodofClassificationofTumorusingCNNUIFigure, 'state');

app.TumorTypeButton.ValueChangedFcn = createCallbackFcn(app, @TumorTypeButtonValueChanged, true);

app.TumorTypeButton.Text = 'Tumor Type';

app.TumorTypeButton.BackgroundColor = [0 0 0];

app.TumorTypeButton.FontColor = [0.9412 0.9412 0.9412];

app.TumorTypeButton.Position = [453 468 156 34];

% Create TumorSpotButton

app.TumorSpotButton = uibutton(app.ANovelMethodofClassificationofTumorusingCNNUIFigure, 'state');

app.TumorSpotButton.ValueChangedFcn = createCallbackFcn(app, @TumorSpotButtonValueChanged, true);

app.TumorSpotButton.Text = 'Tumor Spot';

app.TumorSpotButton.BackgroundColor = [0 0 0];

app.TumorSpotButton.FontColor = [0.9412 0.9412 0.9412];

app.TumorSpotButton.Position = [665 468 156 34];

% Create HomeButton

app.HomeButton = uibutton(app.ANovelMethodofClassificationofTumorusingCNNUIFigure, 'state');

app.HomeButton.ValueChangedFcn = createCallbackFcn(app, @HomeButtonValueChanged, true);

app.HomeButton.Text = 'Home';

app.HomeButton.BackgroundColor = [0 0 0];

app.HomeButton.FontColor = [0.9412 0.9412 0.9412];

app.HomeButton.Position = [87 468 105 34];

% Create usermanual

app.usermanual = uipanel(app.ANovelMethodofClassificationofTumorusingCNNUIFigure);

app.usermanual.BorderType = 'none';

app.usermanual.Position = [72 53 770 374];

% Create Label

app.Label = uilabel(app.usermanual);

app.Label.Position = [19 260 692 48];

app.Label.Text = 'To check the tumor in the MRI scan is of type Benign or Malignant that is Benign is taken as Non-Cancerous and Malignant is ';

% Create Label\_2

app.Label\_2 = uilabel(app.usermanual);

app.Label\_2.Position = [19 226 692 48];

app.Label\_2.Text = 'Cancerous, click the Benign or Malignant Button';

% Create Label\_3

app.Label\_3 = uilabel(app.usermanual);

app.Label\_3.Position = [19 174 721 48];

app.Label\_3.Text = 'To check the tumor in the MRI scan is of which type of tumor, this version is providing this classification with an 87.87% and we are ';

% Create Label\_4

app.Label\_4 = uilabel(app.usermanual);

app.Label\_4.WordWrap = 'on';

app.Label\_4.Position = [19 140 721 48];

app.Label\_4.Text = 'classifying between four types they are Glioma, Pituitary, Meningioma and No Tumor. For this Feature click on the Tumor Type Button';

% Create Label\_5

app.Label\_5 = uilabel(app.usermanual);

app.Label\_5.WordWrap = 'on';

app.Label\_5.Position = [19 67 715 48];

app.Label\_5.Text = {'To get the Tumot Spot click on the Tumor Spot button this feature help you in some extent to highliht the tumor spot '; ' '; 'Click on the image button after the pop up and select the MRI scan u want to process. '};

% Show the figure after all components are created

app.ANovelMethodofClassificationofTumorusingCNNUIFigure.Visible = 'on';

end

end

% App creation and deletion

methods (Access = public)

% Construct app

function app = A\_Novel\_Method\_of\_classification\_of\_Tumor\_Using\_CNN\_exported

% Create UIFigure and components

createComponents(app)

% Register the app with App Designer

registerApp(app, app.ANovelMethodofClassificationofTumorusingCNNUIFigure)

if nargout == 0

clear app

end

end

% Code that executes before app deletion

function delete(app)

% Delete UIFigure when app is deleted

delete(app.ANovelMethodofClassificationofTumorusingCNNUIFigure)

end

end

end