

Додаток А
Програмний код додатку "Object recognition"

```

cnn_v1.mat

mlroot = 'D:\Dataset'

Datasetpath = fullfile(mlroot, '', 'Training')
Data=imageDatastore(Datasetpath, 'IncludeSubfolders', true,
'LabelSource', 'foldernames')
layer = [imageInputLayer([100 100 3])
         convolution2dLayer([5 5],16)
         batchNormalizationLayer
         reluLayer

         maxPooling2dLayer(2, 'Stride',2)

         convolution2dLayer([5 5],32)
         batchNormalizationLayer
         reluLayer

         maxPooling2dLayer(2, 'Stride',2)

         convolution2dLayer([5 5],64)
         batchNormalizationLayer
         reluLayer

         maxPooling2dLayer(2, 'Stride',2)

         fullyConnectedLayer(5)
         softmaxLayer
         classificationLayer];

options = trainingOptions('sgdm', ...
    'InitialLearnRate',0.01, ...
    'MaxEpochs',3, ...
    'Shuffle','every-epoch', ...,
    'ValidationData',Data, ...,
    'Verbose',false, ...
    'Plots','training-progress');

```

```

cnn_v2.mat

```

```

mlroot = 'D:\DS'

Datasetpath = fullfile(mlroot, '', 'Training')
Data=imageDatastore(Datasetpath, 'IncludeSubfolders', true,
'LabelSource', 'foldernames')
layer = [imageInputLayer([100 100 3])
         convolution2dLayer([5 5],16)
         batchNormalizationLayer
         reluLayer

         maxPooling2dLayer(2, 'Stride',2)

```

```

convolution2dLayer([5 5],32)
batchNormalizationLayer
reluLayer

maxPooling2dLayer(2,'Stride',2)

convolution2dLayer([5 5],64)
batchNormalizationLayer
reluLayer

maxPooling2dLayer(2,'Stride',2)

fullyConnectedLayer(4)
softmaxLayer
classificationLayer];

options = trainingOptions('sgdm', ...
    'InitialLearnRate',0.01, ...
    'MaxEpochs',10, ...
    'Shuffle','every-epoch', ...,
    'ValidationData',Data, ...,
    'Verbose',false, ...
    'Plots','training-progress');

main.m

function varargout = main(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',  gui_Singleton, ...
                  'gui_OpeningFcn', @main_OpeningFcn, ...
                  'gui_OutputFcn',  @main_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

handles.output = hObject;

```

```

guidata(hObject, handles);
movegui(gcf, 'center')
close(main);
app;

```

```

close(main);
app2;

```

app.m

```

function varargout = app(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn',  @app_OpeningFcn, ...
                  'gui_OutputFcn',  @app_OutputFcn, ...
                  'gui_LayoutFcn',   [] , ...
                  'gui_Callback',    []);

```

```

if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

```

```

if nargin
    [varargout{1:nargout}] = gui_mainfcn(gui_State,
varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end

```

```

function app_OpeningFcn(hObject, eventdata, handles,
varargin)

```

```

handles.output = hObject;

```

```

guidata(hObject, handles);
movegui(gcf, 'center');
axis off;

```

```
function varargout = app_OutputFcn(hObject, eventdata,
handles)
varargout{1} = handles.output;
```

```
function buttonload_image_Callback(hObject, eventdata,
handles)
[baseFileName, folder] = uigetfile({'*.*',
'*.jpg'}, 'Specify an image file');
```

```
fullImageFileName = fullfile(folder, baseFileName);
im_dis=imread(fullImageFileName);
im_result = imresize(im_dis, [100 100]);
axes(handles.axes1);
imagesc(im_dis);
axis off
```

```
load('D:\cnn_v1.mat');
output = classify(convnet, im_result);
set(handles.predict_image, 'String', output);
```

```
function rb1_Callback(hObject, eventdata, handles)
```

```
if get(handles.rb1, 'Value')
```

```
if
```

```
isequal(get(handles.predict_image, 'String'), get(handles.r
b1, 'String'))
```

```
    msgbox("Знайдений об'ект");
```

```
else
```

```
    msgbox("Не знайдений об'ект");
```

```
end
```

```
end
```

```
function rb2_Callback(hObject, eventdata, handles)
```

```
if get(handles.rb2, 'Value')
```

```

if isequal(get(handles.predict_image,'String'),
get(handles.rb2,'String'))
    msgbox("Знайдений об'ект");

else
    msgbox("Не знайдений об'ект");
end
end

function rb3_Callback(hObject, eventdata, handles)

if get(handles.rb3,'Value')

ifisequal(get(handles.predict_image,'String'),
get(handles.rb3,'String'))
    msgbox("Знайдений об'ект");
else
    msgbox("Не знайдений об'ект");
end

end
function rb4_Callback(hObject, eventdata, handles)
if get(handles.rb4,'Value')

if
isequal(get(handles.predict_image,'String'),get(handles.rb
4,'String'))
msgbox("Знайдений об'ект");
else
msgbox("Не знайдений об'ект");
end
end

function rb4_Callback(hObject, eventdata, handles)
if get(handles.rb4,'Value')
if
isequal(get(handles.predict_image,'String'),get(handles.rb4,'S
tring'))
    msgbox("Знайдений об'ект");
else
    msgbox("Не знайдений об'ект");
end
end
end

```

```

function pushbutton4_Callback(hObject, eventdata, handles)
close(app)
main;

app2.m
function varargout = app2(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn',  @app2_OpeningFcn,
                  ...
                  'gui_OutputFcn',   @app2_OutputFcn, ...
                  'gui_LayoutFcn',   [] , ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargin
    [varargout{1:nargout}] = gui_mainfcn(gui_State,
varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end

function app2_OpeningFcn(hObject, eventdata, handles,
varargin)
handles.output = hObject;
guidata(hObject, handles);
movegui(gcf, 'center');
axis off;

function varargout = app2_OutputFcn(hObject, eventdata,
handles)

```

```

varargout{1} = handles.output;

im_dis1=imread('D:\DS\Training\Apple Braeburn\3_100.jpg');
imshow(im_dis1,'Parent',handles.axes2);
axis off

im_dis2=imread('D:\DS\Training\Apple Golden\3_100.jpg');
imshow(im_dis1,'Parent',handles.axes3);
axis off

im_dis4=imread('D:\DS\Training\Apple Granny
Smith\3_100.jpg');
imshow(im_dis4,'Parent',handles.axes4);
axis off

im_dis5=imread('D:\DS\Training\Apple Red
Delicious\3_100.jpg');
imshow(im_dis5,'Parent',handles.axes5);
axis off

function image_load_Callback(hObject, eventdata, handles)
    [baseFileName, folder] = uigetfile({'*.*',
'*.*jpg'}, 'Specify an image file');
    fullImageFileName = fullfile(folder, baseFileName);
    im_dis=imread(fullImageFileName);
    im_result = imresize(im_dis,[100 100]);
    axes(handles.axes6);
    imagesc(im_dis);
    axis off

load('D:\cnn_v2.mat');
output = classify(cnn,im_result);
set(handles.predict_image,'String',output);

function pushbutton4_Callback(hObject, eventdata, handles)
close(app2)
main

```

Додаток Б
Відомість магістерської атестаційної роботи

