ÓBUDAI EGYETEM

Neurális Hálózatok II.

Intelligens Rendszerek

Gyakorlat

Forrás: Mathworks.com

Konvolúciós neurális hálózatok

- Konvolúciós neurális hálózatok az előző órán tanultaktól némileg eltérnek
- Nincs a rejtett rétegben minden neuron minden bemeneti rétegbeli neuronhoz csatlakoztatva
- Betaníthatók egy jellegzetes képi tartalom detektálására, pl. arc, vagy adott objektum

ÓBUDAI EGYETEM

Matlab R-CNN minta

- Matlab command windiow parancs:
  - TrainRCNNStopSignDetectorExample
- Stop tábla detektálására képes


```
%% Train R-CNN Stop Sign Detector
%
%%
% Load training data and network layers.
load('rcnnStopSigns.mat', 'stopSigns',
'layers')
%%
% Add the image directory to the MATLAB
path.
imDir = fullfile(matlabroot, 'toolbox',
'vision', 'visiondata',...
  'stopSignImages');
addpath(imDir);
```


```
% Tanítási beállítások:
% options = trainingOptions('sgdm', ...
  'MiniBatchSize', 32, ...
  'InitialLearnRate', 1e-6, ...
  'MaxEpochs', 10);

% R-CNN detektor tanítása, időigényes!
rcnn = trainRCNNObjectDetector(stopSigns,
layers, options, 'NegativeOverlapRange',
[0 0.3]);
```


```
Step 1 of 3: Extracting region proposals from 27 training images...done.

Step 2 of 3: Training a neural network to classify objects in training data...

Training on single GPU.
Initializing image normalization.
|=========================================================================================|
|     Epoch    |   Iteration  | Time Elapsed |  Mini-batch  |  Mini-batch  | Base Learning|
|              |              |   (seconds)  |     Loss     |   Accuracy   |     Rate     |
|=========================================================================================|
|            1 |            1 |         0.17 |       0.0080 |      100.00% |     1.00e-06 |
|            3 |           50 |        10.88 |       0.2235 |       90.63% |     1.00e-06 |
|            6 |          100 |        26.58 |       0.0015 |      100.00% |     1.00e-06 |
|            8 |          150 |        47.07 |       0.0016 |      100.00% |     1.00e-06 |
|           10 |          190 |        56.36 |       0.0001 |      100.00% |     1.00e-06 |
|=========================================================================================|

Network training complete.

Step 3 of 3: Training bounding box regression models for each object class...100.00%...done.

R-CNN training complete.
**************************************************************************

Elapsed time is 248.185109 seconds.
```

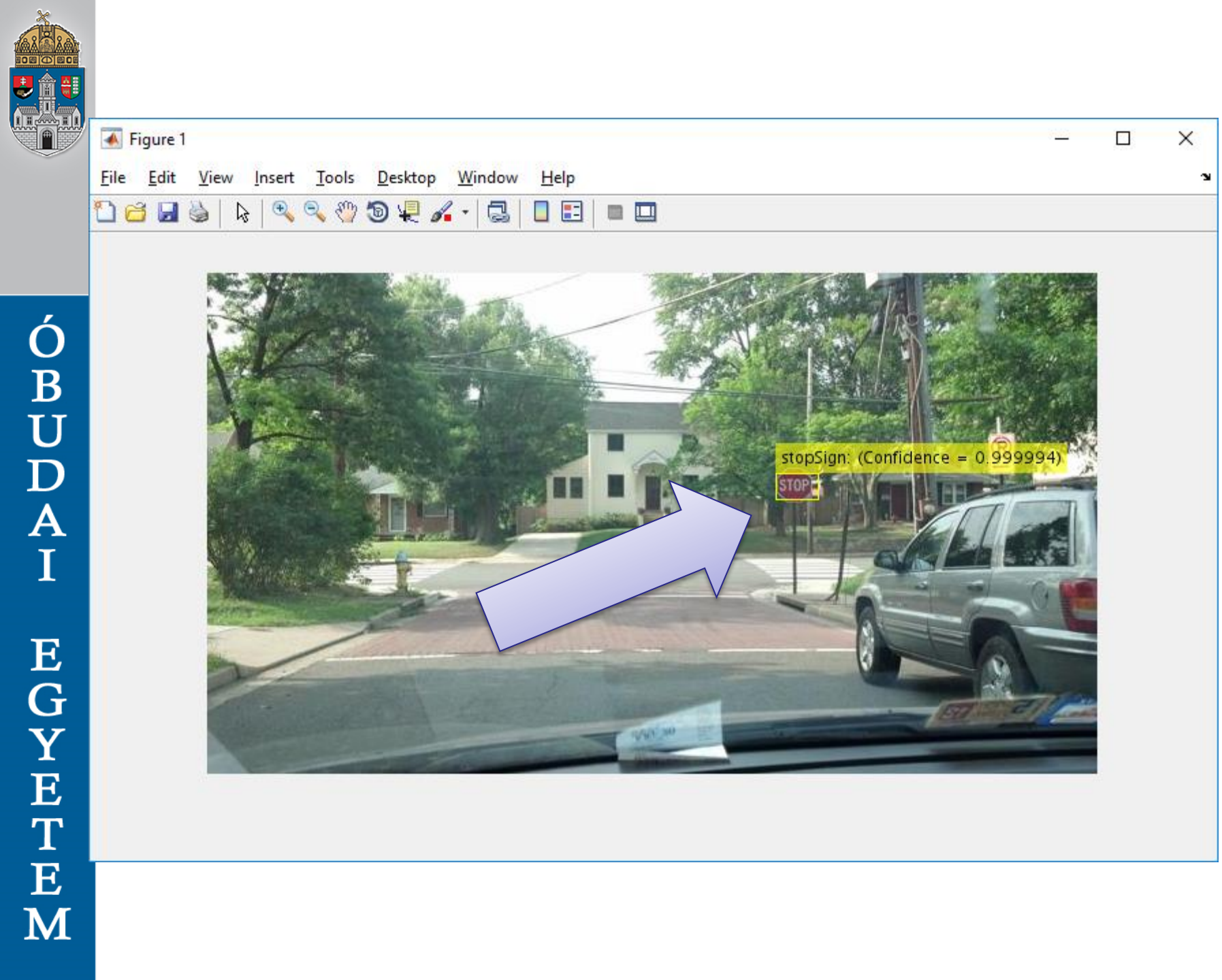

```
% Tesztképen detektálás:
img = imread('stopSignTest.jpg');

[bbox, score, label] = detect(rcnn, img, 'MiniBatchSize',
32);
%
% Legerősebb régió:
[score, idx] = max(score);

bbox = bbox(idx, :);
annotation = sprintf('%s: (Confidence = %f)', label(idx),
score);

detectedImg = insertObjectAnnotation(img, 'rectangle',
bbox, annotation);

% Legerősebb régió ábrázolva:
figure
imshow(detectedImg)
```



ÓBUDAI EGYETEM

Feladat:

Happy/Sad face detektor

R-CNN

- Adott 14db 256x256 pixel felbontású kép
  - 7 boldog
  - 7 szomorú smiley
- A cél betanítani a hálózatot a megfelelő detektálásra
- Kevés a rendelkezésre álló minta!
  - Általában több ezer képre megtanítva működik jól!
- Felosztjuk véletlen 70% tanító és 30% tesztelő mintára

Minták

- Új m file (script)

```
clear

happyImgs = dir('happy*.png');
happy_labels = categorical(repmat(0,numel(happyImgs),1));

sadImgs = dir('sad*.png');
sad_labels = categorical(repmat(1,numel(happyImgs),1));

labels = [happy_labels; sad_labels;];
imds = imageDatastore({'happy*.png'
'sad*.png'},'Labels',labels);

figure
numImages = length(imds.Files);
perm = randperm(numImages,length(imds.Files));
for i = 1:length(imds.Files)
    subplot(2,length(imds.Files)/2,i);
    imshow(imds.Files{perm(i)});
end
```


```
[imdsTrain,imdsTest] =
splitEachLabel(imds,0.7,'randomize');

layers = [ ...
    imageInputLayer([256 256 3])
    convolution2dLayer(5,20)
    reluLayer
    maxPooling2dLayer(2,'Stride',2)
    fullyConnectedLayer(2)
    softmaxLayer
    classificationLayer];

options = trainingOptions('sgdm', ...
    'MaxEpochs',20,...
    'InitialLearnRate',1e-4, ...
    'Verbose',0, ...
    'Plots','training-progress');

net = trainNetwork(imdsTrain,layers,options);
```


```
YPred = classify(net,imdsTest);
YTest = imdsTest.Labels;

accuracy = sum(YPred == YTest)/numel(YTest)

%Eredmény 100%-os tanítás esetén:

accuracy =

     1
```