

UNIVERZITET U BIHAĆU
TEHNIČKI FAKULTET
BIHAĆ

Auditorne vježbe iz predmeta
VJEŠTAČKA INTELIGENCIJA I EKSPERTNI SISTEMI

Una Drakulić, MA elektrotehnike
Viši asistent

DUBOKO UČENJE (DEEP LEARNING)

Konvolucijska neuronska mreža (CNN) je poseban tip duboke mreže koja je efikasna za obradu 2D podataka (slike). Ključne osobine:

- **Konvolucijski slojevi** (filteri) automatski uče lokalne obrasce (rubove, teksture).
- **Pooling slojevi** smanjuju dimenzije i čuvaju značajke.
- **Fully connected i softmax** slojevi mapiraju izdvojene značajke na klase.

MATLAB pruža visoke apstrakcije (layers, datastores, trainingOptions, trainNetwork) koje ubrzavaju razvoj i eksperimentiranje.

Učitavanje i priprema podataka

imageDatastore — čita slike iz foldera i automatski pravi label-e iz imena foldera.

```
imds = imageDatastore('path/to/data', 'IncludeSubfolders', true,  
'LabelSource', 'foldernames');
```

splitEachLabel — podjela po klasama (train/val/test):

```
[imdsTrain, imdsVal, imdsTest] = splitEachLabel(imds, 0.7, 0.15, 0.15,  
'randomize');
```

augmentedImageDatastore — skaliranje/augmentacija pri učitavanju (korisno za pretrained mreže koje očekuju specifične dimenzije):

```
augImdsTrain = augmentedImageDatastore([224 224], imdsTrain,  
'ColorPreprocessing', 'gray2rgb');
```

imageDataAugmenter — definiranje augmentacija (rotacija, flip, shift):

```
augmenter = imageDataAugmenter('RandRotation', [-10 10], 'RandXTranslation', [-3  
3], 'RandYReflection', true);  
augImds = augmentedImageDatastore([28 28], imdsTrain, 'DataAugmentation',  
augmenter);
```

Primjer1

Dataset sadrži slike rukom pisanih cifara 0–9. **imageDatastore** je poseban MATLAB objekat koji čita slike iz foldera i automatski im dodjeljuje oznake (labels) prema nazivu foldera. Svaka slika će biti povezana s klasom (0–9) prema folderu u kojem se nalazi.

```
close all  
clear all  
clc  
  
%ucitavanje slika, svaka slika je gray skala 28x28 piksela  
digitDatasetPath = fullfile(matlabroot, 'toolbox', 'nnet', 'nndemos', ...  
    'nndatasets', 'DigitDataset');  
imds = imageDatastore(digitDatasetPath, ...  
    'IncludeSubfolders', true, 'LabelSource', 'foldernames');  
  
% test/train/validation  
% 60% trenje, 20%validacija i 20% testiranje  
fracTrainFiles = 0.6;  
fracValFiles = 0.2;  
fracTestFiles = 0.2;
```

```

[imdsTrain, imdsValidation, imdsTest] = splitEachLabel(imds, ...
    fracTrainFiles, fracValFiles, fracTestFiles, 'randomize');

% definiranje CNN arhitekture
%Struktura:
%Input layer: ulaz 28×28×1 (grayscale slika).
%Conv layer: 10 filtera, kernel 3×3, padding 'same'.
%Batch norm: stabilizira učenje.
%Max pooling: smanjuje dimenzije slike (downsampling).
%Drugi conv + ReLU: dodaje nelinearnost i složenije reprezentacije.
%Fully connected layer (10): mapira na 10 klasa (cifre 0-9).
%Softmax: pretvara rezultate u vjerovatnoće.
%Classification layer: koristi cross-entropy gubitak.

layers = [
    imageInputLayer([28 28 1])

    convolution2dLayer(3,10,'Padding','same')
    batchNormalizationLayer
    maxPooling2dLayer(2,'Stride',2)

    convolution2dLayer(3,10,'Padding','same')
    batchNormalizationLayer
    reluLayer

    fullyConnectedLayer(10)
    softmaxLayer
    classificationLayer];

%% Stochastic Gradient Descent with Momentum (SGDM)
% Početna brzina učenja: 0.01
% 100 epoha
% Validacija svakih 30 iteracija
% Prikazuje training-progress graf

options = trainingOptions('sgdm', ...
    'InitialLearnRate',0.01, ...
    'MaxEpochs',100, ...
    'Shuffle','every-epoch', ...
    'ValidationData',imdsValidation, ...
    'ValidationFrequency',30, ...
    'Verbose',false, ...
    'Plots','training-progress');

net = trainNetwork(imdsTrain,layers,options);

%%

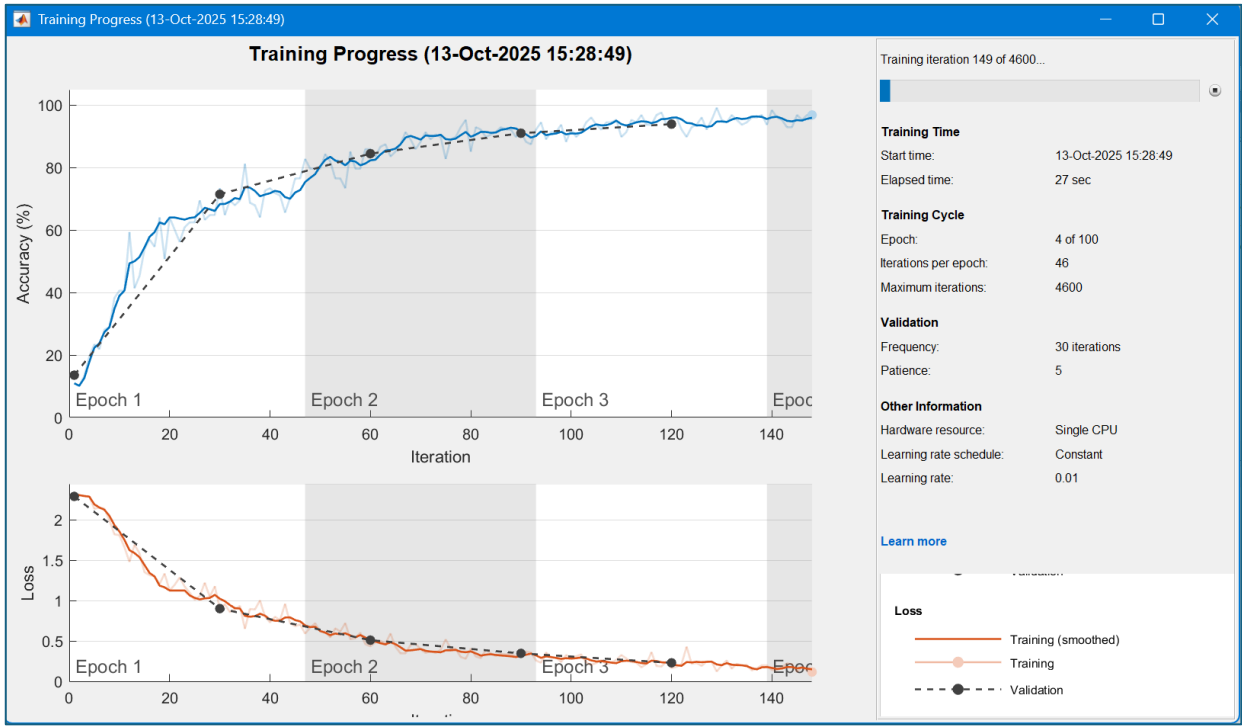
YPred = classify(net,imdsTest);
YTest = imdsTest.Labels;

% broj tacno klasificiranih/boj test uzoraka
accuracy = sum(YPred == YTest)/numel(YTest)

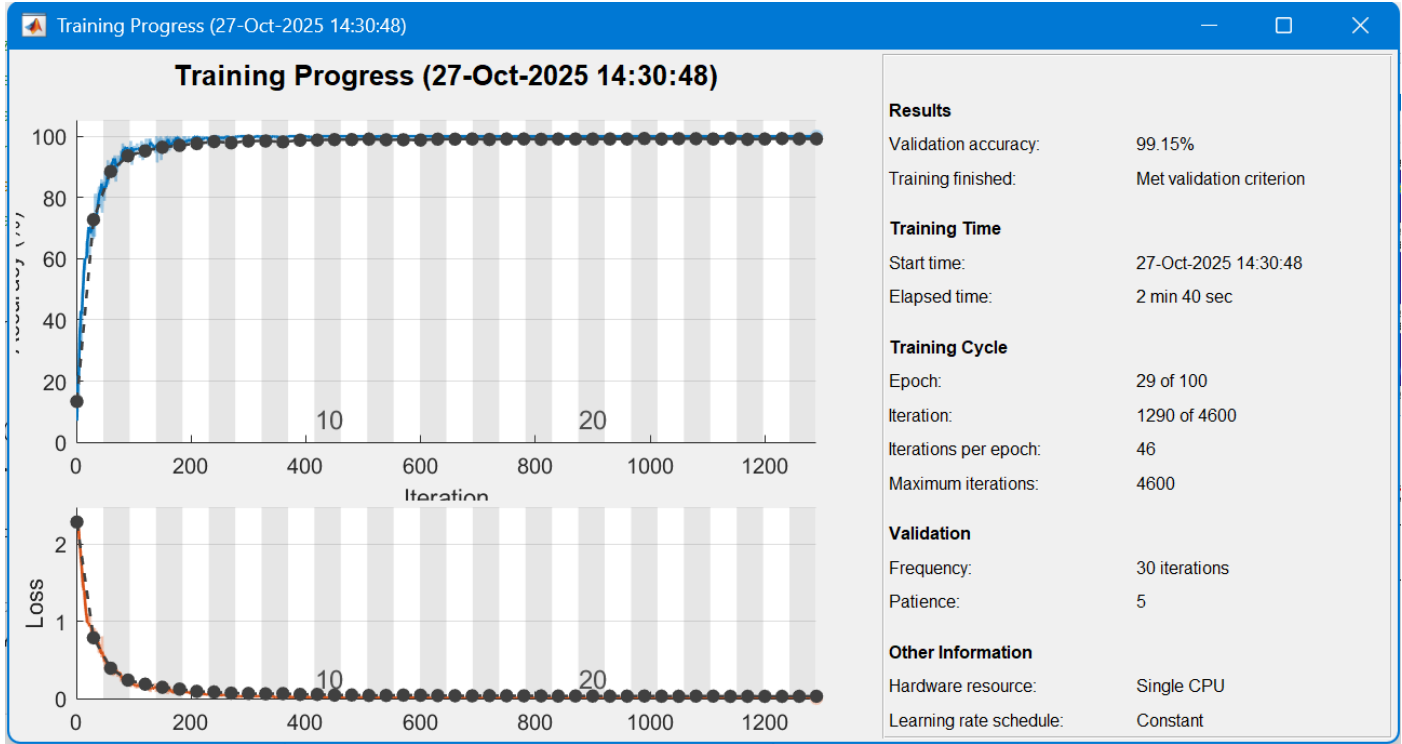
%%
ind = find(YPred ~= YTest);
figure;
for ii = 1:9
    subplot(3,3,ii);
    imagesc(imdsValidation.readimage(ind(ii)));

```

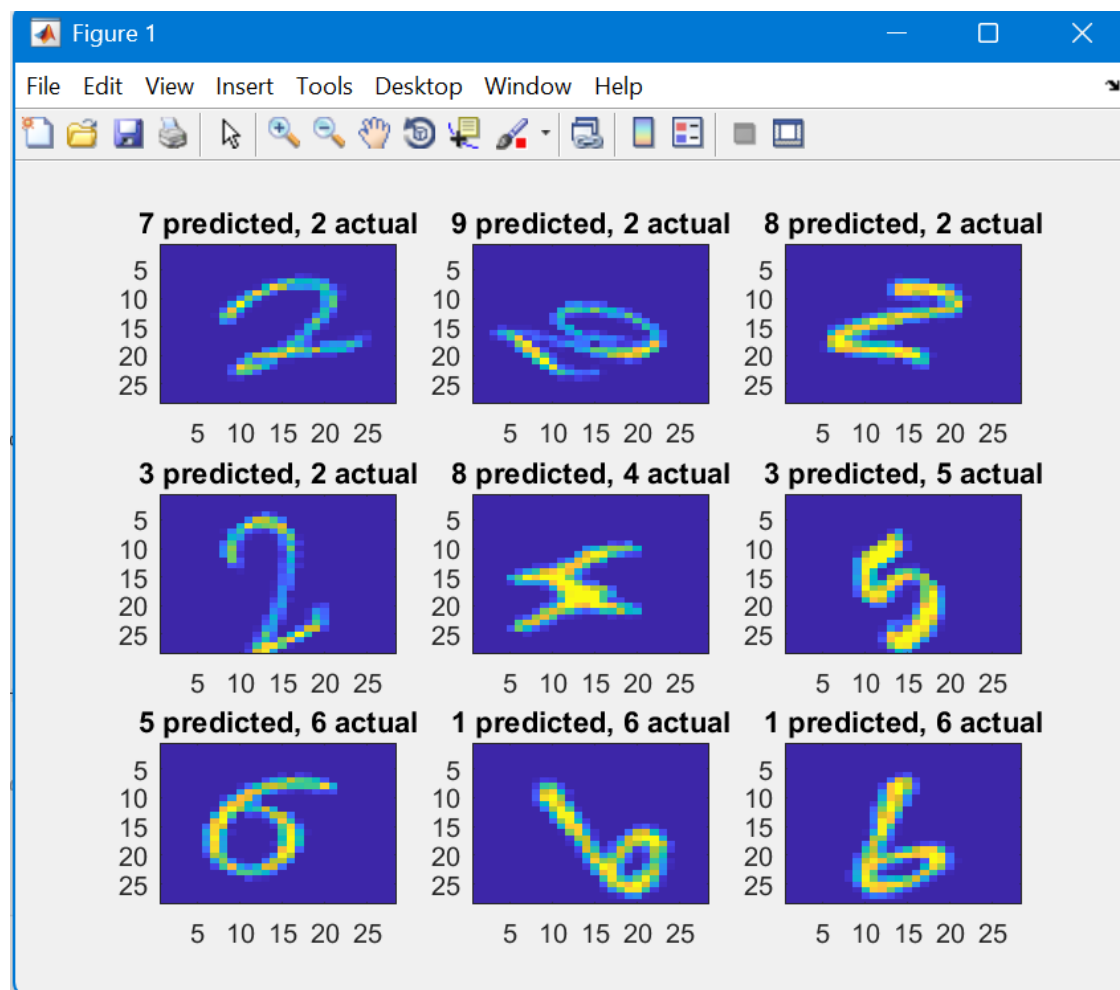
```
title([num2str(double(YPred(ind(ii))))-1), ' predicted, ', ...
      num2str(double(YTest(ind(ii))))-1, ' actual'])
end
```



Slika 1



Slika 2



Slika 3

Primjer2

Potrebno je napisati program koji prepoznaje znakove sa registarske tablice vozila primjenom metoda obrade slike i poređenja šablona (template matching). Potrebno je implementirati kod koji učitava sliku vozila sa vidljivom registarskom tablicom, izdvaja područje registarske tablice iz slike korištenjem morfoloških operacija i detekcije ivica, razdvaja pojedinačne karaktere (slova i brojeve) sa tablice, prepoznaje svaki karakter poređenjem sa unaprijed definisanim šablonima i prikazuje prepoznatu registarsku oznaku na ekranu.

Rješenje:

Skripta	Funkcija	Šta radi
Create_templates.m	Kreira bazu poznatih znakova	Učitava slike slova/brojeva, sprema ih u .mat datoteku
readLetter.m	Prepoznaje jedan znak	Poredi ulaznu sliku znaka sa bazom pomoću korelacije
numberplate.m	Glavni program	Izdvađa tablicu sa slike, prepoznaje znakove, sklapa registarski broj

Create_templates.m kreira bazu poznatih slika (slova i brojeva) koje se koriste kao referenca za poređenje.

Create_templates.m

```
%CREATE TEMPLATES
%Letter
A=imread('char/A.bmp');B=imread('char/B.bmp');
C=imread('char/C.bmp');D=imread('char/D.bmp');
E=imread('char/E.bmp');F=imread('char/F.bmp');
G=imread('char/G.bmp');H=imread('char/H.bmp');
I=imread('char/I.bmp');J=imread('char/J.bmp');
K=imread('char/K.bmp');L=imread('char/L.bmp');
M=imread('char/M.bmp');N=imread('char/N.bmp');
O=imread('char/O.bmp');P=imread('char/P.bmp');
Q=imread('char/Q.bmp');R=imread('char/R.bmp');
S=imread('char/S.bmp');T=imread('char/T.bmp');
U=imread('char/U.bmp');V=imread('char/V.bmp');
W=imread('char/W.bmp');X=imread('char/X.bmp');
Y=imread('char/Y.bmp');Z=imread('char/Z.bmp');
Afill=imread('char/fillA.bmp');
Bfill=imread('char/fillB.bmp');
Dfill=imread('char/fillD.bmp');
Ofill=imread('char/fillO.bmp');
Pfill=imread('char/fillP.bmp');
Qfill=imread('char/fillQ.bmp');
Rfill=imread('char/fillR.bmp');

%iz foldera char učitavaju se slike svih karaktera A-Z i 0-9 u formatu .bmp

%Number
one=imread('char/1.bmp'); two=imread('char/2.bmp');
three=imread('char/3.bmp');four=imread('char/4.bmp');
five=imread('char/5.bmp'); six=imread('char/6.bmp');
seven=imread('char/7.bmp');eight=imread('char/8.bmp');
nine=imread('char/9.bmp'); zero=imread('char/0.bmp');
zerofill=imread('char/fill0.bmp');
fourfill=imread('char/fill4.bmp');
sixfill=imread('char/fill6.bmp');
sixfill2=imread('char/fill6_2.bmp');
eightfill=imread('char/fill8.bmp');
ninefill=imread('char/fill9.bmp');
ninefill2=imread('char/fill9_2.bmp');

%Creating the array of alphabets
letter=[A Afill B Bfill C D Dfill E F G H I J K L M N O Ofill P Pfill Q Qfill R Rfill S T U V W X Y Z];
%Creating the array of numbers sve slike se spajaju u jedan niz
number=[one two three four fourfill five six sixfill sixfill2 seven eight eightfill nine ninefill ninefill2
zero zerofill];

character=[letter number];
% svaka ćelija cell sadrži jednu sliku veličine 42x24 piksela
NewTemplates=mat2cell(character,42,[24 24 24 24 24 24 24 24 ...
```

```
save ('NewTemplates','NewTemplates')
clear all
```

readLetter.m

```
function letter=readLetter(snap)
%READLETTER reads the character fromthe character's binary image.
% LETTER=READLETTER(SNAP) outputs the character in class 'char' from the
% input binary image SNAP.

load NewTemplates % Loads the templates of characters in the memory.
snap=imresize(snap,[42 24]); % Resize the input image so it can be compared with the template's
images.
comp=[ ];
for n=1:length(NewTemplates)
% corr2 racuna koeficijent korelacije izmedu slike znaka i svakog od šablona
%vrijednost ide -1 1, sto je veca bliza je 1, slike su slicnije.
    sem=corr2(NewTemplates{1,n},snap); % Correlation the input image with every image in the
template for best matching.
    comp=[comp sem]; % Record the value of correlation for each template's character.
    %display(sem);
end

%vd je indeks znaka iz baze koji najvise lici na ulazni znak
vd=find(comp==max(comp)); % Find the index which correspond to the highest matched character.
%display(max(comp));
%*-_*-*_*-*_*-*_*-*_*-
% Accodrding to the index assign to 'letter'.
% Alphabets listings.
% if elseif blok određuje koji karakter odgovara tom indeksu jer vise šablona može biti isti znak
%na kraju funkcije treba da vraca prepoznati znak u obliku stringa npr. 'A' '5' ...
if vd==1 || vd==2
    letter='A';
elseif vd==3 || vd==4
    letter='B';
elseif vd==5
    letter='C';
elseif vd==6 || vd==7
    letter='D';
elseif vd==8
    letter='E';
elseif vd==9
```

```
letter='F';
elseif vd==10
    letter='G';
elseif vd==11
    letter='H';
elseif vd==12
    letter='I';
elseif vd==13
    letter='J';
elseif vd==14
    letter='K';
elseif vd==15
    letter='L';
elseif vd==16
    letter='M';
elseif vd==17
    letter='N';
elseif vd==18 || vd==19
    letter='O';
elseif vd==20 || vd==21
    letter='P';
elseif vd==22 || vd==23
    letter='Q';
elseif vd==24 || vd==25
    letter='R';
elseif vd==26
    letter='S';
elseif vd==27
    letter='T';
elseif vd==28
    letter='U';
elseif vd==29
    letter='V';
elseif vd==30
    letter='W';
elseif vd==31
    letter='X';
elseif vd==32
    letter='Y';
elseif vd==33
    letter='Z';
    %*_*_*_*_*
% Numerals listings.
elseif vd==34
    letter='1';
elseif vd==35
    letter='2';
elseif vd==36
    letter='3';
elseif vd==37 || vd==38
    letter='4';
elseif vd==39
```



```
letter='5';
elseif vd==40 || vd==41 || vd==42
    letter='6';
elseif vd==43
    letter='7';
elseif vd==44 || vd==45
    letter='8';
elseif vd==46 || vd==47 || vd==48
    letter='9';
else
    letter='0';
end
end
```

Sljedeći m-file ima za cilj da izdvoji i prepozna registarsku tablicu sa slike automobila/vozila.

numberplate.m

```
clc;
close all;
clear all;

%ucitavanje slike
%im = imread('Images/audi.jpg');
im = imread('Images/6.jpg');
%im = imread('Images/car1.PNG');

%resize slike
im = imresize(im, [480 NaN]);

%prikaz originalne slike
imshow(im),title("Originalna reg. tabla");

%pretvori u grayscale
imgray = rgb2gray(im);

%pretvori u binarnu
imbin = imbinarize(imgray);

%detekcija rubova korištenjem Sobelovog algoritma
im = edge(imgray, 'sobel');

%proširuje sliku koristeći dijamantsku strukturu
im = imdilate(im, strel('diamond', 2));

%fills holes by making it full white color
im = imfill(im, 'holes');

%extract the solid filled parts
im = imerode(im, strel('diamond', 10));

%calculates Area,Bounding Box and Image size
Iprops=regionprops(im,'BoundingBox','Area', 'Image');

%stores the area of the first element
```

```

area = Iprops.Area;
%counts the number of elements
count = numel(Iprops);
%variable to store max area, initial value=first element's area
maxa= area;
%stores bounding box
boundingBox = Iprops.BoundingBox;
%loop to find the bounding box with the maximum area
for i=1:count
    if maxa<Iprops(i).Area
        maxa=Iprops(i).Area;
        boundingBox=Iprops(i).BoundingBox;
    end
end

%all above steps are to find location of the number plate
%Now crop the binarized image to get the number plate only
im = imcrop(imbin, boundingBox);

%resize number plate to 240 NaN
im = imresize(im, [240 NaN]);

%clear dust
im = imopen(im, strel('rectangle', [4 4]));

%remove some object if it's width is too long or too small than 500
im = bwareaopen(~im, 500);

%get width
[h, w] = size(im);

figure;
imshow(im),title("Izdvojena reg. tabla sa izolovanim karakterima");

% Read letter
Iprops=regionprops(im,'BoundingBox','Area', 'Image');
count = numel(Iprops);

noPlate=[]; % Initializing the variable of number plate string.

hold on; % this is used to currunt plot active fro green box
for i=1:count % this fo loop is used to give the green border to
extracted number plate
    ow = length(Iprops(i).Image(1,:));
    oh = length(Iprops(i).Image(:,1));
    if ow<(h/2) && oh>(h/3)
        rectangle('Position', Iprops(i).BoundingBox, 'EdgeColor', 'g',
'LineWidth', 2);
    end
end
for i=1:count
    %width of the ith character of number plate
    ow = length(Iprops(i).Image(1,:));
    %length of the ith character of number plate
    oh = length(Iprops(i).Image(:,1));
    if ow<(h/2) && oh>(h/3)

```

```

letter=readLetter(Iprops(i).Image); % Reading the letter
corresponding the binary image 'N'.
figure; imshow(Iprops(i).Image);
noPlate=[noPlate letter] % Appending every subsequent character
in noPlate variable.
end
end

```

Registracijska tabla je: APO9 BN 7886

