



Computer Vision

Classification with Neural Networks

10 April 2018

Copyright © 2001 – 2018 by
NHL Stenden Hogeschool and Van de Loosdrecht Machine Vision BV
All rights reserved

j.van.de.loosdrecht@nhl.nl, jaap@vdlmv.nl

Classification with Neural Networks

Overview:

- Introduction classification
- Introduction neural networks
- Image classification
- Exercise
- Feature vector classification (*)
- Exercise (*)

Related presentations:

- Classification with neural networks part II
with examples how to use neural networks in scripts
- Genetic Algorithms
how to tune the parameters of a neural networks

27-Aug-18

Classification with Neural
Networks

2

Introduction classification

Classification:

the assignment of an object characterized by a set of features to one of a number of predefined classes.

Example: Optical Character Recognition

27-Aug-18

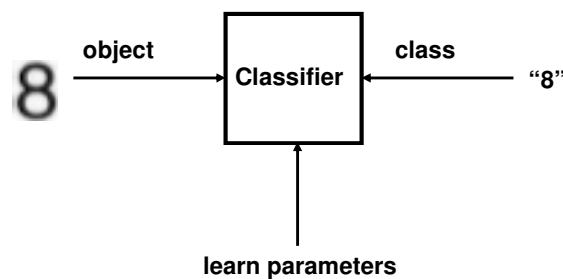
Classification with Neural
Networks

3

Introduction classification

- **Learning mode**

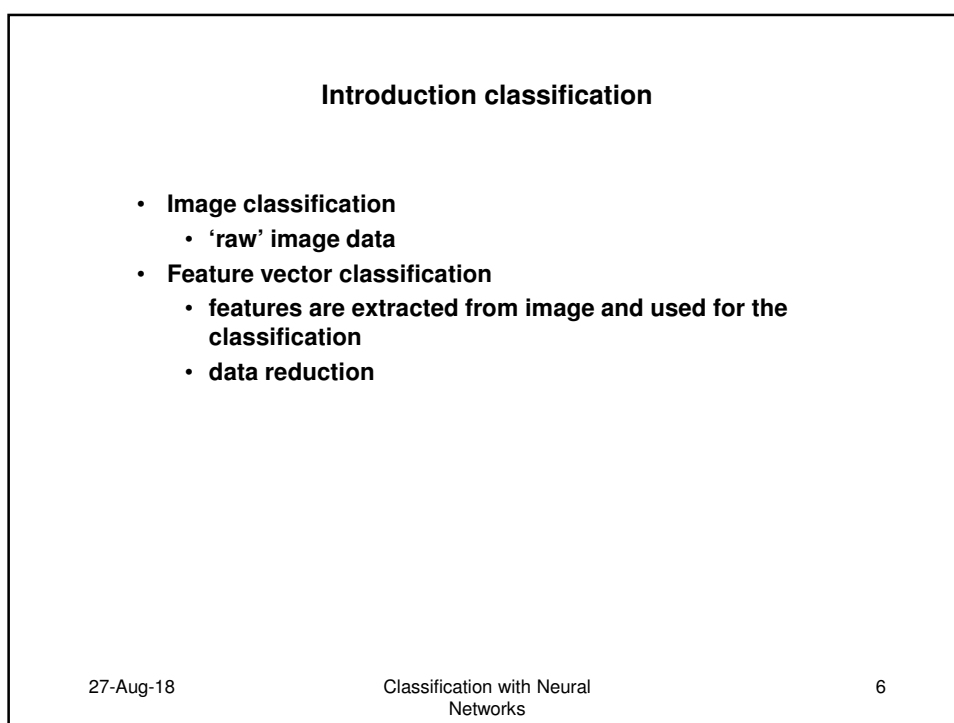
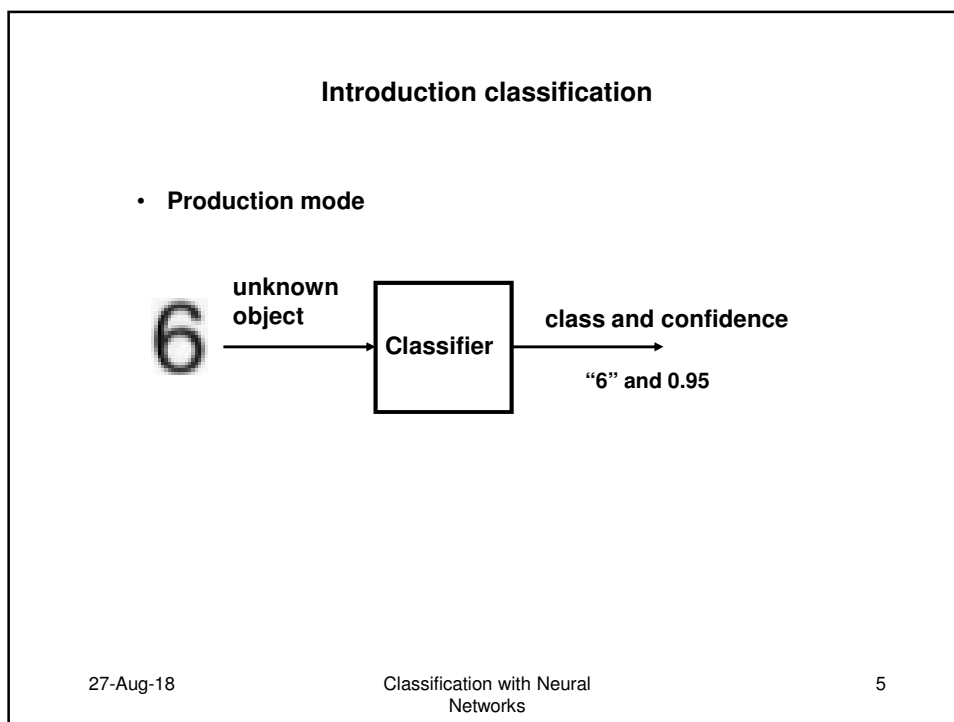
- **Learn set: set of objects with known classes**
- **Each time the total learn set is trained is called an epoch, for successful learn normally many epochs are necessary**



27-Aug-18

Classification with Neural
Networks

4



Introduction neural networks

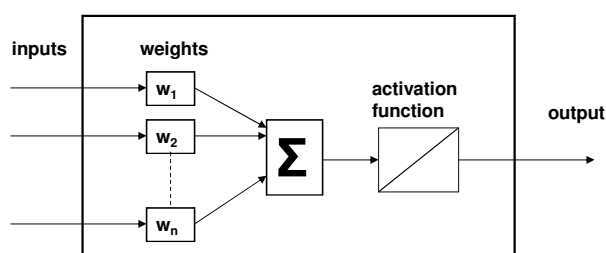
- Based on a model of our brain
- Neuron
- Neural networks
- Multilayer perceptron neural Network with back propagation learning
- Example: learning sinus function

27-Aug-18

Classification with Neural
Networks

7

Neuron



27-Aug-18

Classification with Neural
Networks

8

Neural networks

- Neural network consists of multiple neurons organised in different layers
- There are several different architectures for the organization and for the learning algorithms
 - Supervised learning
 - Unsupervised learning
- Most common used architecture is a multilayer perceptron neural Network with back propagation learning also called Back Propagation Network (BPN)

27-Aug-18

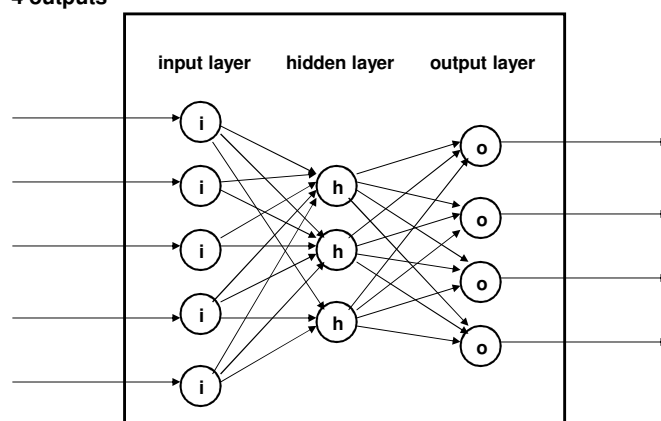
Classification with Neural
Networks

9

Back Propagation Network

Example with:

- 5 inputs
- one hidden layer with 3 neurons
- 4 outputs



27-Aug-18

Classification with Neural
Networks

10

Example: learning sinus function

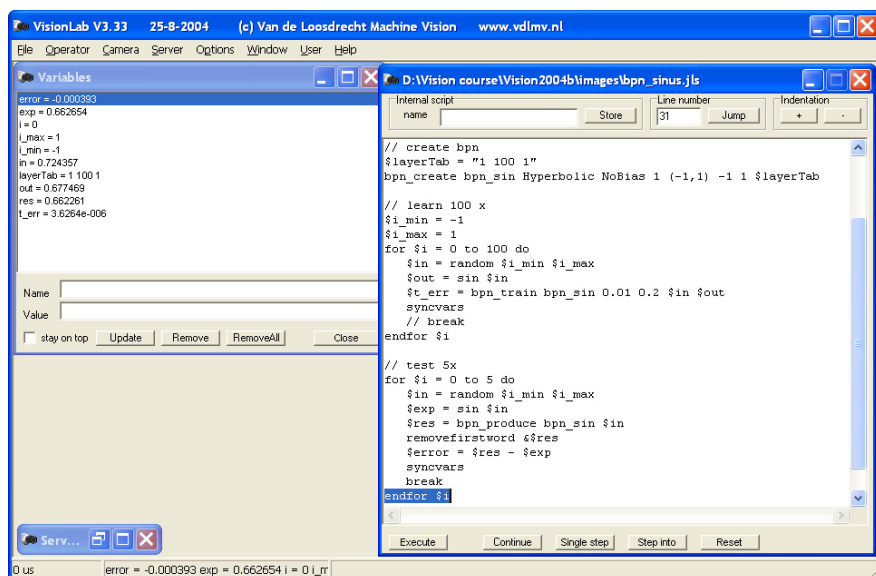
- Create BPN with
 - Input layer of 1 neuron
 - One hidden layer of 100 neurons
 - Output layer of 1 neuron
- Train the BPN with 100 random examples: $x, \sin(x)$
 - \$in = random -1 1
 - \$out = sin \$in
 - Note the behaviour of the training error (\$t_err)
- Test the BPN with 5 random values
 - \$in = random -1 1
 - \$res = sinus of \$in learned by BPN
 - \$exp = sin \$in
 - \$error = error in result
- use script bpn_sinus.jls

27-Aug-18

Classification with Neural
Networks

11

Example: learning sinus function



27-Aug-18

Classification with Neural
Networks

12

Image classification

- Introduction
- Class Image Sets (CIS)
 - Train set
 - Evaluation sets
- Training the BPN
- Using the BPN

27-Aug-18

Classification with Neural
Networks

13

Image classification

- Pixels of image are inputs,
nr of pixels = size input layer
- Nr of classes = size output layer
- Scale, rotation and lighting variations
 - Train BPN with all combinations
 - Use computer vision operations to make images invariant
by normalizing:
 - Scale, use Zoom
 - Rotation, use Rotate(Full)
 - Lighting, use Contrast Stretch

27-Aug-18

Classification with Neural
Networks

14

Class Image Set (CIS)

- A CIS is a collection of images with their associated classes. For usage with Neural Networks, all images in a CIS must have the same image type and size
- CIS:
 - Class table with the name and id of each class
 - For each class an image table, each image in an image table has its unique image index number

27-Aug-18

Classification with Neural
Networks

15

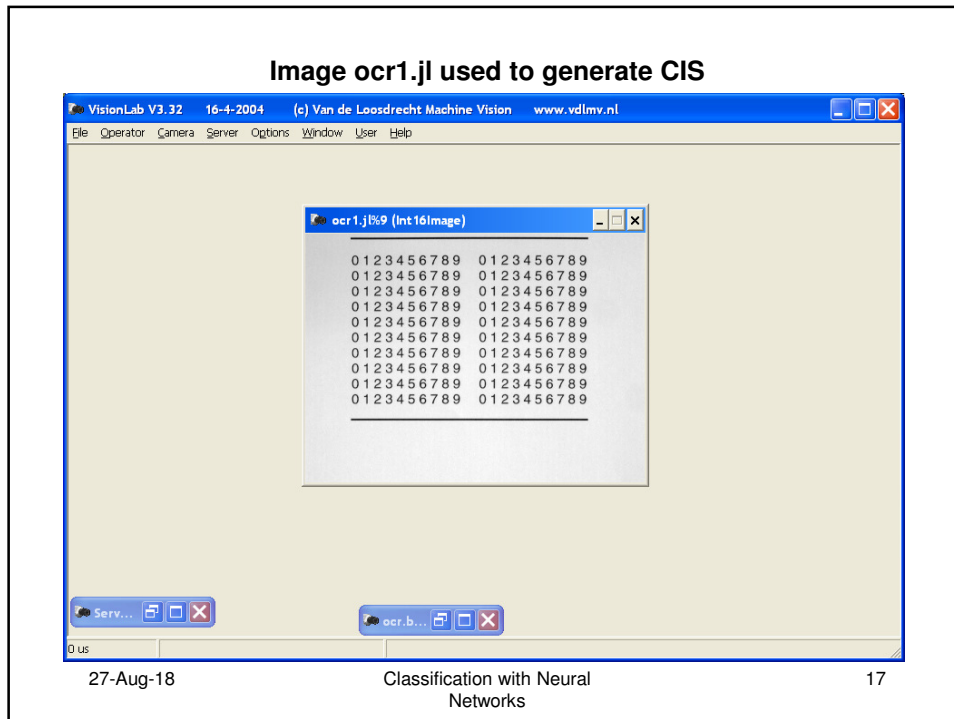
Demonstration Class Image Set (CIS) (*)

- Create new CIS and add some classes and images (no slides)
use correct image type and image size

27-Aug-18

Classification with Neural
Networks

16



Script gen_ocr1_cis.jls

- **Create cis**
- **Threshold image**
- **Remove big blobs**
- **Label image**
- **BlobAnalyse SortDown TopLeft UseX Height TopLeft Width**
- **for \$group = 0 to 1 do**
 - **for \$num = 0 to 9 do**
 - **for \$i = 0 to 9 do**
 - **Roi \$t \$l \$h \$w**
 - **Contraststretch**
 - **Zoom NearestPixelInterpolation**
 - **Add normalized image to cis**

27-Aug-18 Classification with Neural Networks 18

Demonstration Class Image Set (CIS)

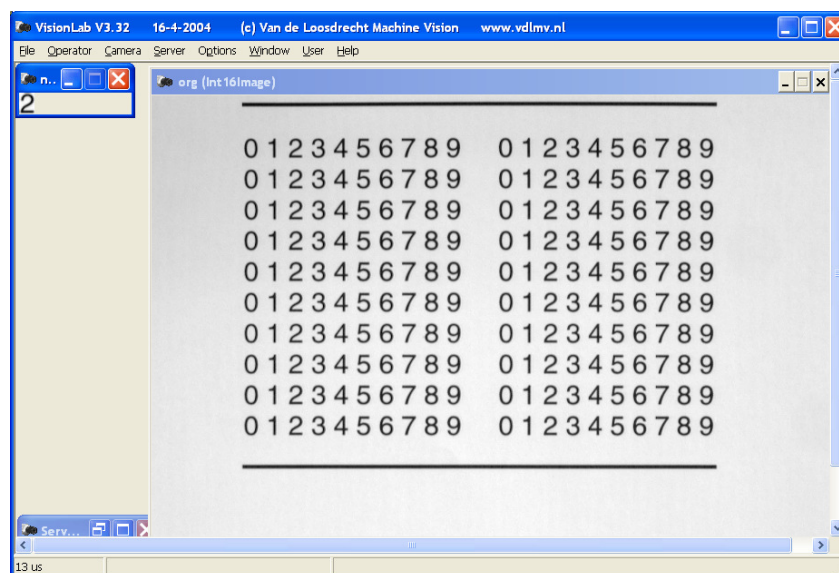
- Use script `gen_ocr1_cis.jls` to generate `ocr1.cis`,
- Open and examine `ocr1.cis`

27-Aug-18

Classification with Neural
Networks

19

Generation of ocr.cis



27-Aug-18

Classification with Neural
Networks

20

ocr1.cis

27-Aug-18

Classification with Neural Networks

21

ocr1.cis

27-Aug-18

Classification with Neural Networks

22

Different sizes of numbers

- **Note that number 1 is much smaller in width as number 8**
- **A BPN always works with a fixed number of inputs.**
- **So all numbers have to be rescaled in size to the same height and width of the learn image**

27-Aug-18

Classification with Neural
Networks

23

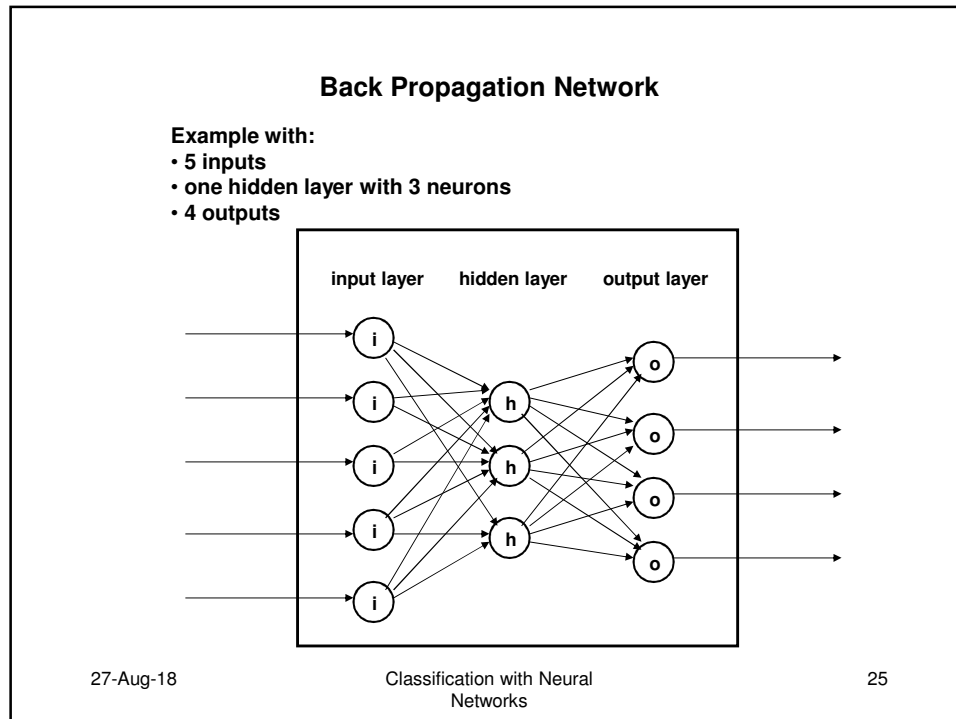
Training the BPN

- **Learning parameters**
- **Train set
balancing the train set**
- **Evaluation set**
- **Testing the result**

27-Aug-18

Classification with Neural
Networks

24



Learning parameters

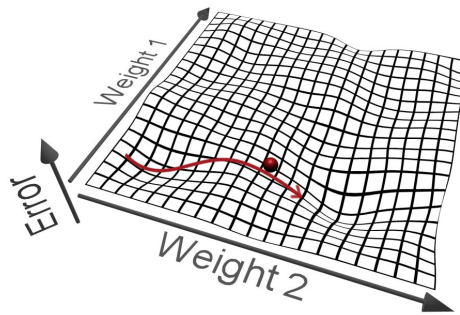
- **Learn rate:**
The learn rate is a small positive value and determines how fast the BPN will learn. If the learn rate is too low learning will be very slow. If the learn rate is too high the training error will oscillate and stay at a high value.
- **Momentum:**
The momentum is zero or a small positive value. This parameter influences the speed of convergence of the learning process and specifies which fraction of the "learn effect" of the previous training sample should be added to the "learn effect" of the next training sample.

27-Aug-18

Classification with Neural Networks

26

Learn rate and momentum



the direction of the change for the weights is calculated using the derivative of the error function

the 'step size' for the adaptation of the weights is proportional to the learning rate

the momentum is the fraction of the previous weight change which is added to the new calculated weight change

27-Aug-18

Classification with Neural
Networks

27

Learning parameters

- **Size first hidden layer:**
If too small the training error will remain high, if too high training will be very slow
- **Size second hidden layer:**
The use of a second hidden layer may speed up the learning process. Advice: start first with an empty 2nd hidden layer.

27-Aug-18

Classification with Neural
Networks

28

Training the BPN

- **Stopping the training process:**
 - Max nr of epochs
 - Stop error
- **Confidence:**
Indication how confident the classifier is about the result,
[1 = confident .. 0 = unconfident]
In normal conditions a confidence > 0.7 is fine
- **Reset network:**
all weights are random initialised with value [min ..max]

27-Aug-18

Classification with Neural
Networks

29

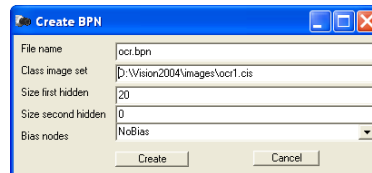
Demonstration Training BPN

- Create new BPN ocr.bpn (File menu),
use ocr1.cis and size first hidden layer = 20, second = 0, NoBias
- Go to Training form, learnRate = 0.001, momentum = 0.001
- Start training, note evaluation set and training set are the same
reset: -0.1 .. 0.1
- Open ocr1.cis and select from a random class a random image and extract that image with
name test
- Select image with name test as "second selected"
- Go to BPN and select Classify form and classify "second selected" image

27-Aug-18

Classification with Neural
Networks

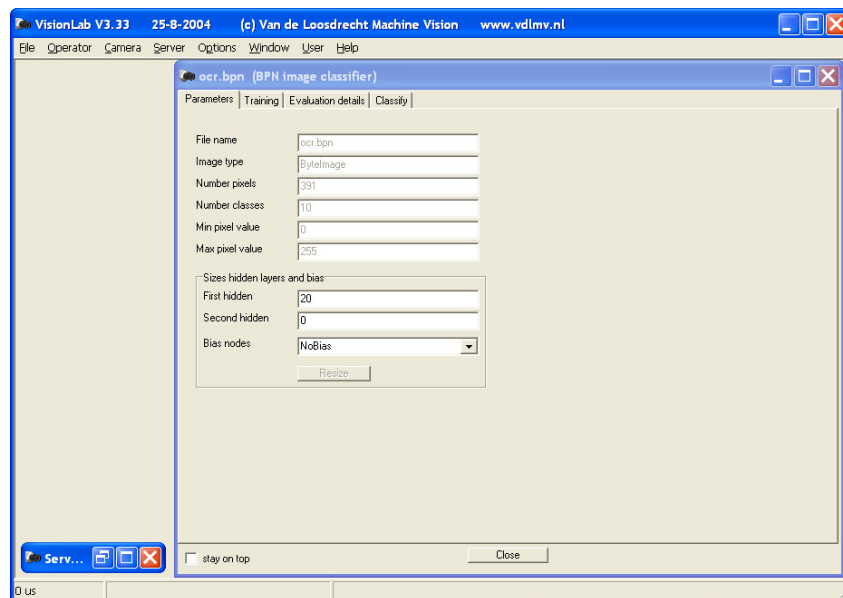
30

Demonstration Training BPN

27-Aug-18

Classification with Neural
Networks

31

Demonstration Training BPN

27-Aug-18

Classification with Neural
Networks

32

Demonstration Training BPN

VisionLab V3.3325-8-2004(c) Van de Loosdrecht Machine Visionwww.vdlnv.nl

FileOperatorCameraServerOptionsWindowUserHelp

ocr.bpn (BPN image classifier)

ParametersTrainingEvaluation detailsClassify

Training parameters

Train setD:\Vision2004\images\ocr1.cisNew

Learn rate0.001Momentum0.001

Max nr epochs1000Stop error0

Evaluation parameters

Evaluation setD:\Vision2004\images\ocr1.cisNew

Evaluation epochs1Low confidence0.7

Training results

Current epoch292

Mean error0.000137919

Maximum error0.00118737

Evaluation results

Mean error0.000124561Maximum error0.00105333

Miss classifications0Low confidences0

History

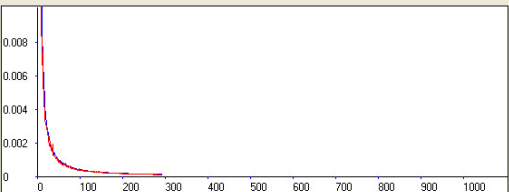
Mean errors

Training

Evaluation

max error0.01

Update



StartStopResetmin0.1max0.1

12.096 us0.001053330.0001245610.0

27-Aug-18

Classification with Neural Networks

33

Select random image from random class

VisionLab V3.3325-8-2004(c) Van de Loosdrecht Machine Visionwww.vdlnv.nl

FileOperatorCameraServerOptionsWindowUserHelp

ocr1.cis (Class image set of ByteImage)

File nameD:\Vision2004\images\ocr1.cis

Infobig number test

Image height23Image width17

name	id	nr images
eight	8	20
five	5	20
four	4	20
nine	9	20
one	1	20
seven	7	20
six	6	20
three	3	20
two	2	20

Image table




Image index5

Add imageRemove image

Extract image test

Min Max pixel value0255

Add class

Remove class

stay on top

Close

7

0 us

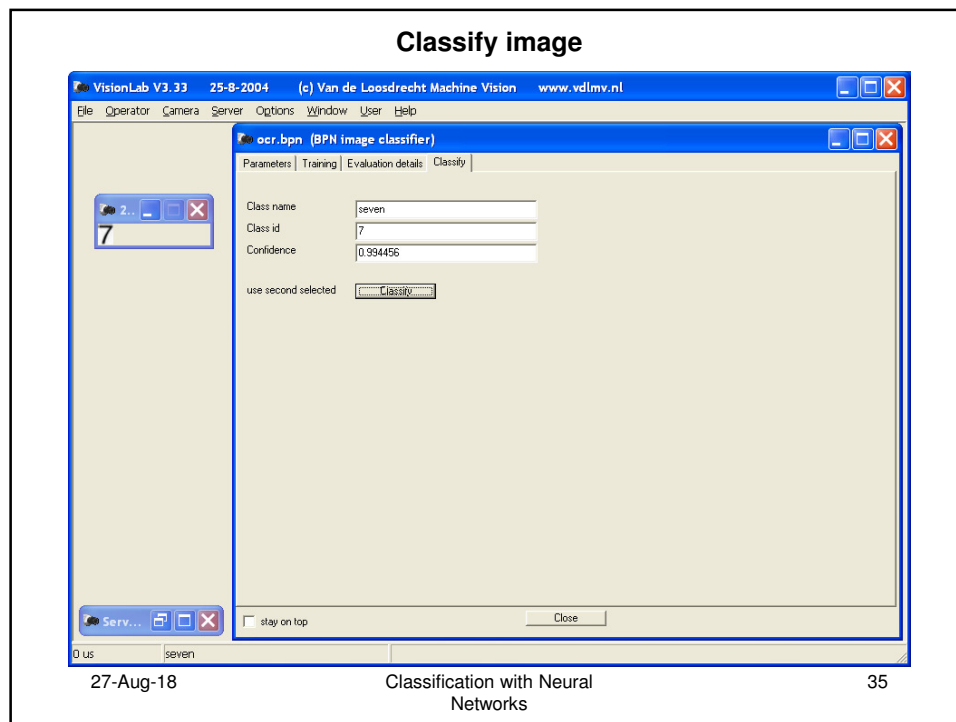
27-Aug-18

Classification with Neural Networks

34

Jaap van de Loosdrecht, NHL, vdLMV, j.van.de.loosdrecht@nhl.nl

17



Training with evaluation set

In order to avoid inbreed, it's better to use a separate training and evaluation set

Demonstration Training BPN with evaluation set

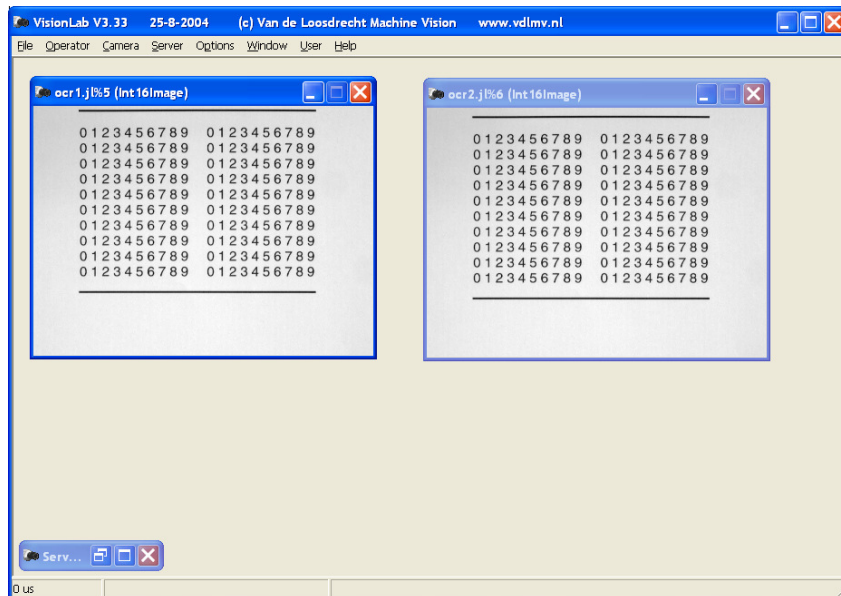
- Create new CIS and some classes (no slides)
- Open image ocr1.jl and ocr2.jl, same object slightly different position and lighting condition
- Use script gen_ocr2_cis.jls to generate ocr2.cis
- Create new BPN ocr2.bpn (File menu), use ocr1.cis and size first hidden layer = 20, second = 0, NoBias
- Go to Training form
- Start training, and select as evaluation set ocr2.cis, learnRate = 0.001, momentum = 0.001
- Open ocr2.cis and select from a random class a random image and extract that image with name test
- Select image with name test as "second selected"
- Go to BPN and select Classify form and classify "second selected" image

27-Aug-18

Classification with Neural
Networks

37

Compare image ocr1.jl and ocr2.jl

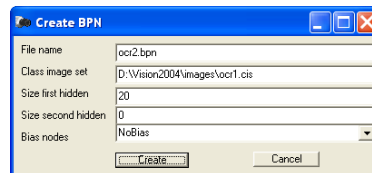


27-Aug-18

Classification with Neural
Networks

38

Demonstration Training BPN with evaluation set

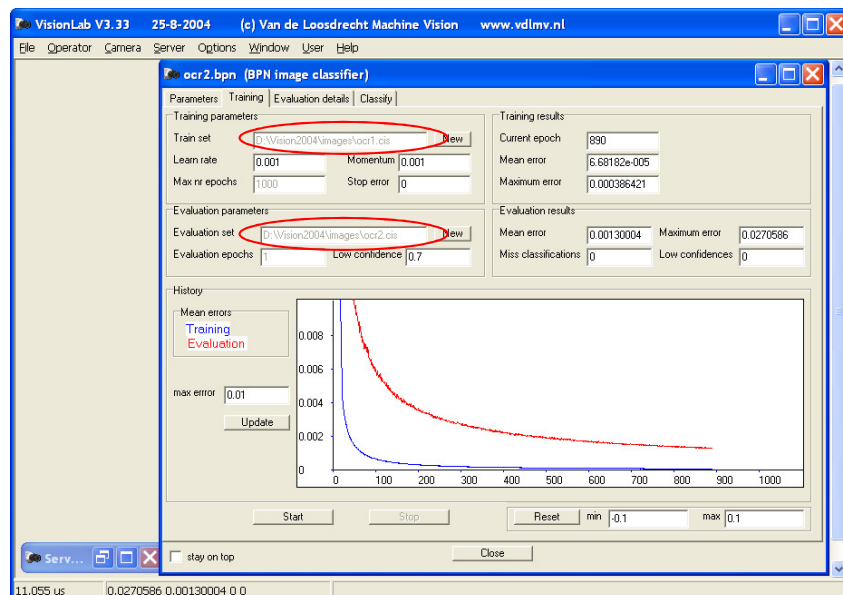


27-Aug-18

Classification with Neural
Networks

39

Demonstration Training BPN with evaluation set

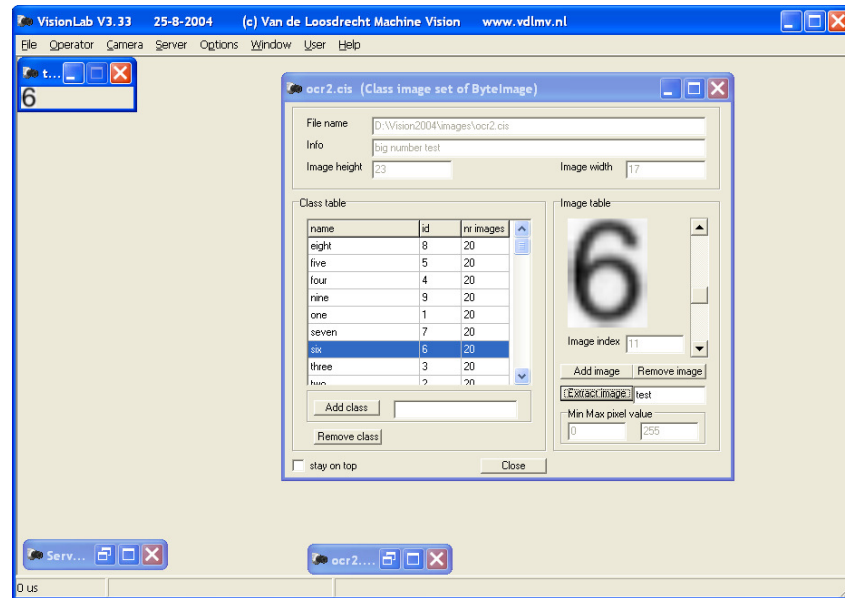


27-Aug-18

Classification with Neural
Networks

40

Select random image from random class from evaluation CIS

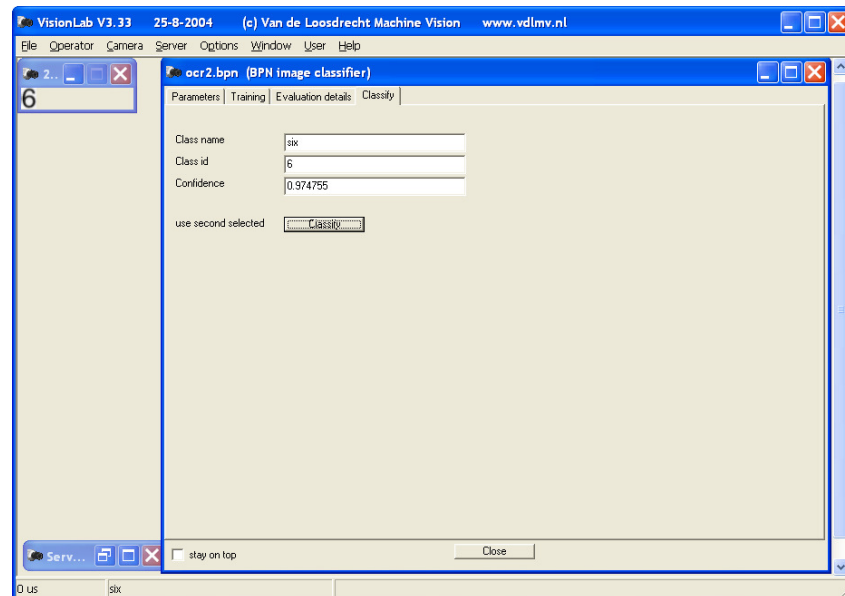


27-Aug-18

Classification with Neural
Networks

41

Classify image



27-Aug-18

Classification with Neural
Networks

42

Optimizing the settings for a image BPN

This function uses a Genetical Algorithm (GA) in order to find the best solution.

NOTE: this optimalization process is very time consuming and for real problems it may take hours or even days.

In the Low/High box the user must specify the search space for size of hidden layers, learn rate and momentum, by specifying the lowest and highest border values.

In the settings box must be specified:

- the population size for the GA algorithm
- the number of epochs the BPN is trained before being evaluated
- the low confidency factor used in the evaluation
- the maximum number of generation for the GA algorithm

27-Aug-18

Classification with Neural
Networks

43

Optimizing the settings for a image BPN

In the detailed setting box, specialized setting can be set, it is advised to use the standard values.

- the MinError is the minimal target error
- the DeltaError is the minimal error for start of hill climbing
- the MicroP is the probability during self mate that parameter is a micro mutation

With the Start button the optimising process is started. With the Stop button the optimising process is stopped, but first the current generation is optimised completely, note this can take some time!

In the best result box the best values found sofar for hidden layers, learn rate and momentum are displayed. Also the current generation number and the best error value found.

With the Accept button the best solution found is used to initialise the settings in the Parameters and Training tab. Then the normal full training is started.

27-Aug-18

Classification with Neural
Networks

44

Demonstration genetic algorithm for optimizing image BPN

- Go to GA Optimize form, this can be used to find the optimal setting for the BPN
Use settings:
Low and High first hidden = 20 (is fixed for the demo)
Low and High second hidden = 0
LearnRate: 0.0005 – 0.001
Momentum: 0 – 0.001
Populationsize: 4
NrEpochs: 10
LowConfidence: 0.5
MinError: 0
DeltaError: 1
MircoP: 0.25
- Click start to start optimise process
Note this process is very time consuming, for the demo PopulationSize and NrEpochs have been set to low values. Also the first hidden layers is fixed to 10
- Click stop button to interrupt the optimise process
- Click accept button to initialise the settings in the Parameters and Training tab. Then the normal full training is started.

27-Aug-18

Classification with Neural
Networks

45

Set the parameters for the optimising process

Low/High	
Low	High
First hidden: 10	10
Second hidden: 0	0
Learn rate: 0.0005	0.001
Momentum: 0	0.001

Best results	
10	Generation Nr: 10
0	Error: 0.041988
0.00064892	
4.55048e-005	

Settings	
Population size: 4	
Nr Epochs: 20	
Low confidence: 0.5	
Max generations: 100	

Detailed settings	
Min Error: 0	
Delta Error: 1	
MicroP: 0.25	

Start Stop Accept

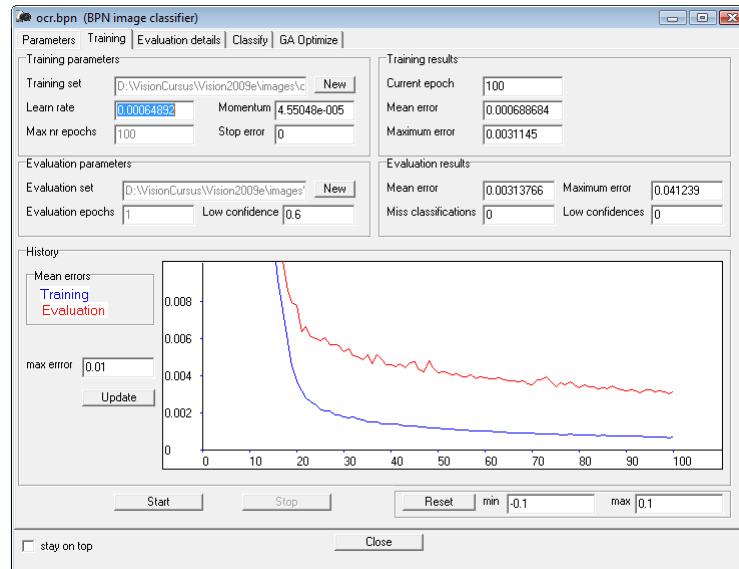
☐ stay on top Close

27-Aug-18

Classification with Neural
Networks

46

Used find optimal parameters to start the full training



27-Aug-18

Classification with Neural
Networks

47

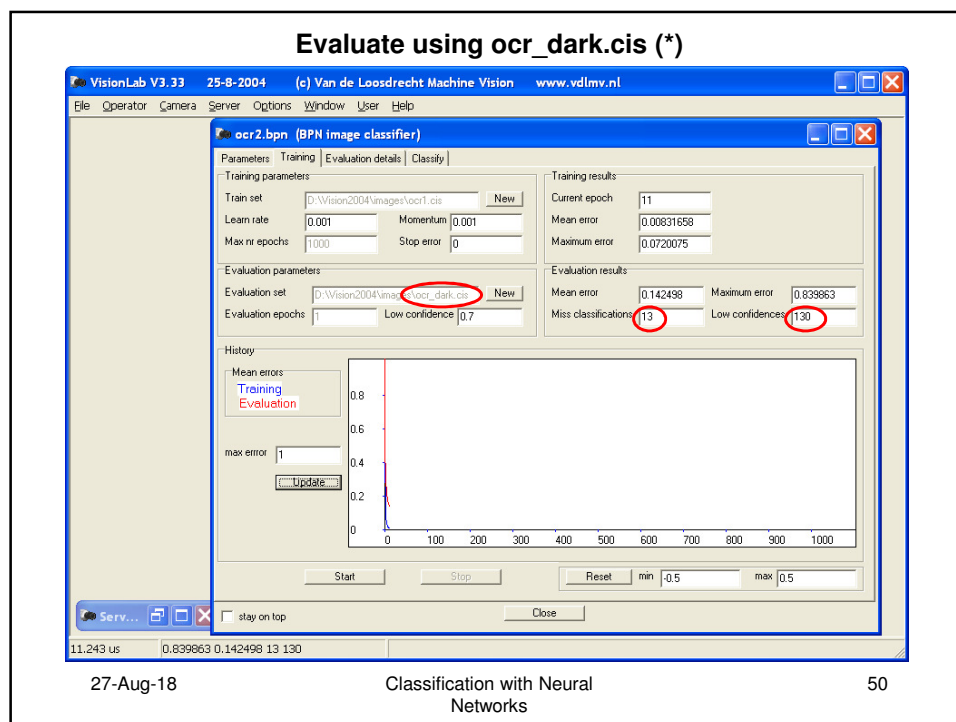
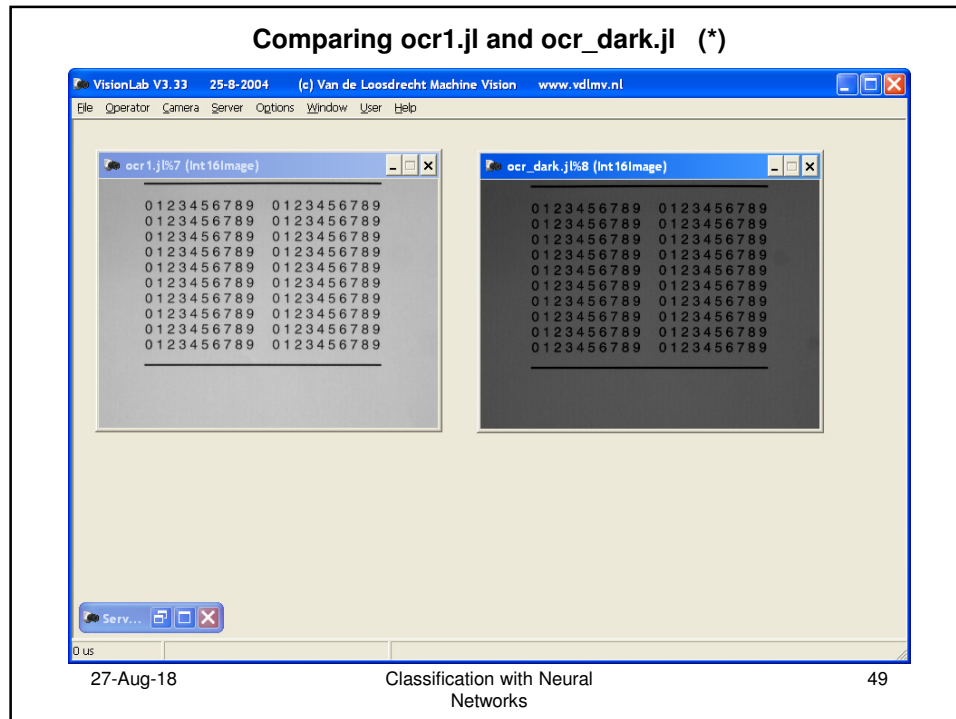
Demo analyzing learning problems (*)

- Compare images ocr1.jl and ocr_dark.jl with [LUT clip](#)
- Change evaluation set to ocr_dark.cis, this is generated from image ocr_dark.jl under very low lighting conditions
- **Reset network with -0.5 and 0.5**, ([-0.1 .. 0.1] will give better results)
- Start training and stop after 12 epochs, due to the low number of epochs there are now 11 miss classifications and 99 low confidences
- This can be analyzed on the Evaluation details form, use low details
- By selecting a row and specifying a name at the Extract Image button the offending image can be extracted for examination
- By changing to high details the individual outputs in the output layer can be examined. Output values are in the range [-1..1]. A class is 'strong' if the output value is close to 1 and weak if close to -1.
On the first row an 'Eight' is classified as a 'Three'

27-Aug-18

Classification with Neural
Networks

48



Evaluation details form (*)

expected	result	confidence	image index	zero	one	two
three	eight	0.582345	5	0.231293	0.202556	-0.00959396
three	eight	0.259483	13	0.340741	0.160064	-0.012493
three	eight	0.269721	16	0.263056	0.20315	-0.0274982
six	two	0.157174	11	0.184081	0.0455188	0.637648
six	two	0.0298297	14	0.159878	0.120233	0.609769
six	two	0.0721116	17	0.189499	0.256491	0.648581
nine	two	0.0866289	2	0.619824	-0.284899	0.737399
nine	zero	0.0389737	3	0.54314	-0.262549	0.502391
nine	zero	0.0895142	4	0.882921	-0.231134	0.73784
nine	zero	0.176234	6	0.5895	-0.247549	0.410051
nine	two	0.105652	11	0.669756	-0.229718	0.827996
nine	zero	0.0391194	12	0.895932	-0.230957	0.828529
nine	two	0.111245	17	0.647347	-0.227412	0.809403
zero	zero	0.648314	0	0.951215	0.0826692	0.00749155
one	one	0.614229	1	0.263941	1.10444	-0.0168601
one	one	0.593233	2	0.274405	1.0748	-0.0281802
one	one	0.637681	3	0.243149	1.09904	-0.0285751
one	one	0.601265	4	0.273329	1.09765	0.000801291
one	one	0.562173	5	0.302555	1.07952	-0.00918783

27-Aug-18

Classification with Neural
Networks

51

Extract offending image (*)

expected	result	confidence	image index	zero	one	two
three	eight	0.582345	5	0.231293	0.202556	-0.00959396
three	eight	0.259483	13	0.340741	0.160064	-0.012493
three	eight	0.269721	16	0.263056	0.20315	-0.0274982
six	two	0.157174	11	0.184081	0.0455188	0.637648
six	two	0.0298297	14	0.159878	0.120233	0.609769
six	two	0.0721116	17	0.189499	0.256491	0.648581
nine	two	0.0866289	2	0.619824	-0.284899	0.737399
nine	zero	0.0389737	3	0.54314	-0.262549	0.502391
nine	zero	0.0895142	4	0.882921	-0.231134	0.73784
nine	zero	0.176234	6	0.5895	-0.247549	0.410051
nine	two	0.105652	11	0.669756	-0.229718	0.827996
nine	zero	0.0391194	12	0.895932	-0.230957	0.828529
nine	two	0.111245	17	0.647347	-0.227412	0.809403
zero	zero	0.648314	0	0.951215	0.0826692	0.00749155
one	one	0.614229	1	0.263941	1.10444	-0.0168601
one	one	0.593233	2	0.274405	1.0748	-0.0281802
one	one	0.637681	3	0.243149	1.09904	-0.0285751
one	one	0.601265	4	0.273329	1.09765	0.000801291
one	one	0.562173	5	0.302555	1.07952	-0.00918783

27-Aug-18

Classification with Neural
Networks

52

Examine high details (*)

Miss classification
3 classified as 8
3: 0.1.3766
8: 0.876285

Low confidences
0: 0.263941
1: 1.10444

27-Aug-18

Classification with Neural Networks

53

Examine high details, scrolled to the right (*)

Miss classification
3 classified as 8
3: 0.1.3766
8: 0.876285

Low confidences
0: 0.263941
1: 1.10444

27-Aug-18

Classification with Neural Networks

54

Exercise OCR dark (*)

- **Experiment with learning parameters to get better results**

27-Aug-18

Classification with Neural
Networks

55

Notes for applications

- **It's better to control the lighting conditions so this kind of dark image can not occur**
- **It's also possible to use more sophisticated contrast normalization algorithms to deal with changing light conditions**
- **The training set can be extended with images under different lighting conditions to make the application more robust**

27-Aug-18

Classification with Neural
Networks

56

Demo robustness of BPN

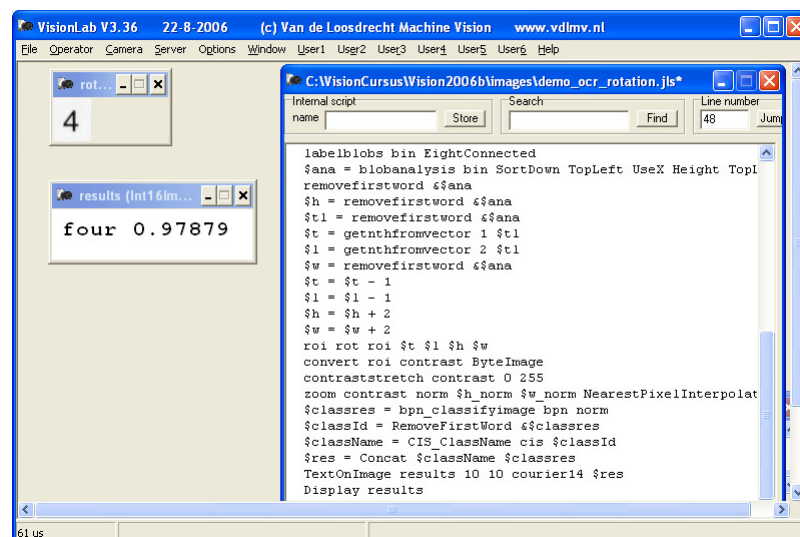
- Execute script demo_ocr_rotation.jls
- Execute script demo_ocr_size.jls
- Execute script demo_ocr_contrast.jls

27-Aug-18

Classification with Neural
Networks

57

Demo with rotated input images

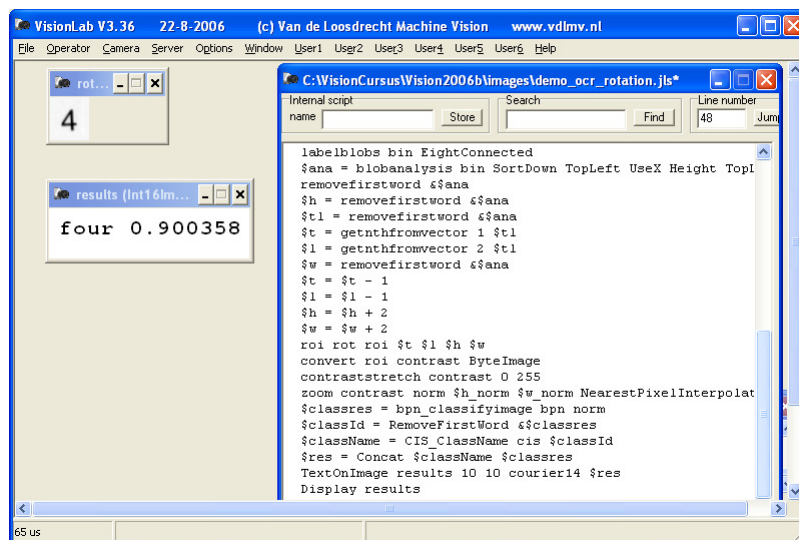


27-Aug-18

Classification with Neural
Networks

58

Demo with rotated input images

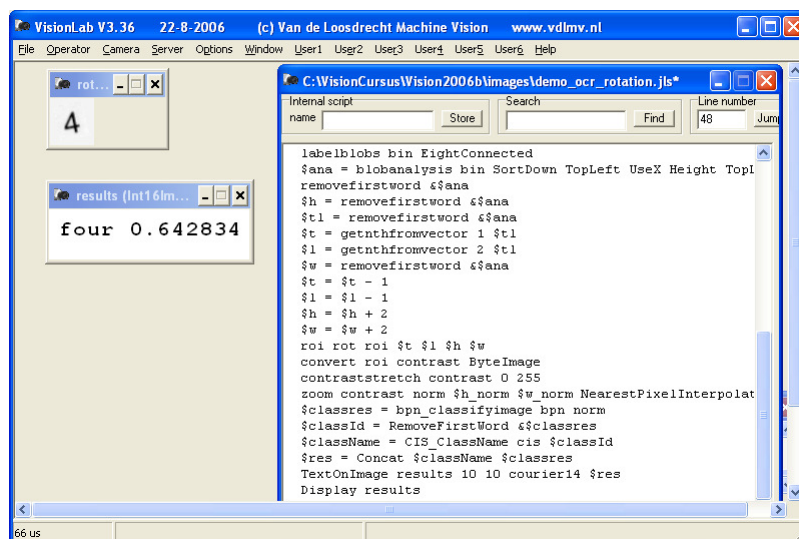


27-Aug-18

Classification with Neural
Networks

59

Demo with rotated input images

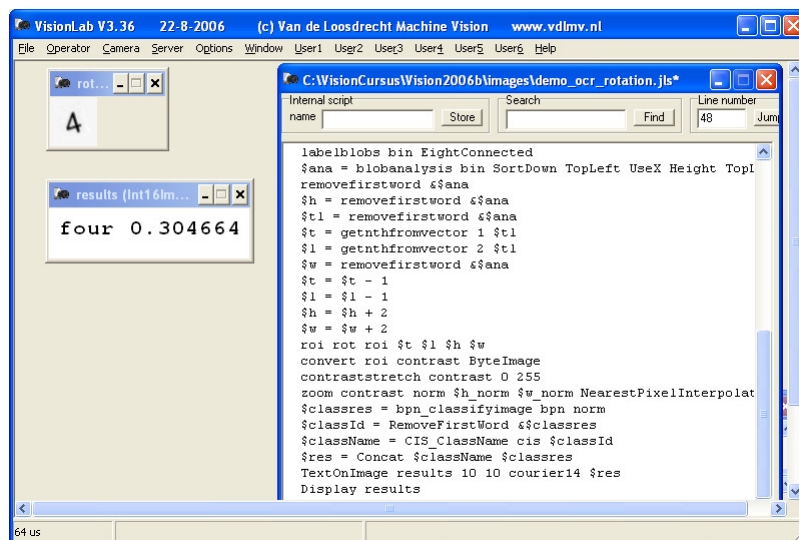


27-Aug-18

Classification with Neural
Networks

60

Demo with rotated input images

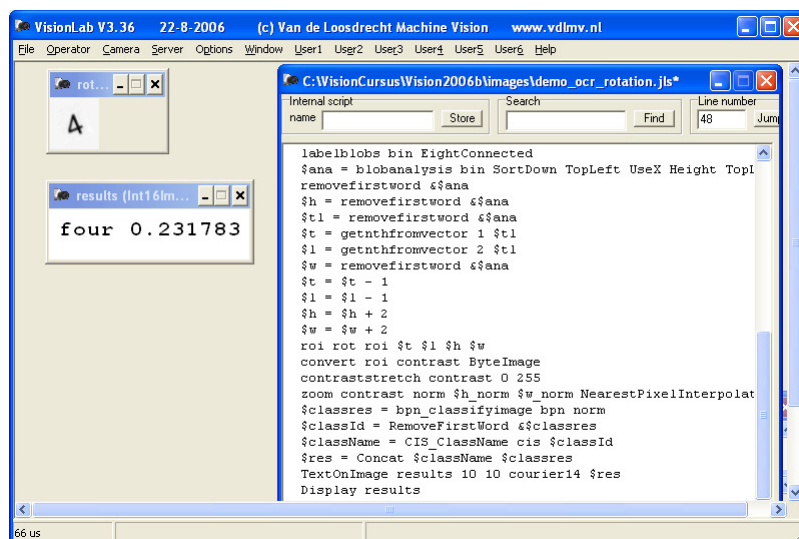


27-Aug-18

Classification with Neural
Networks

61

Demo with rotated input images

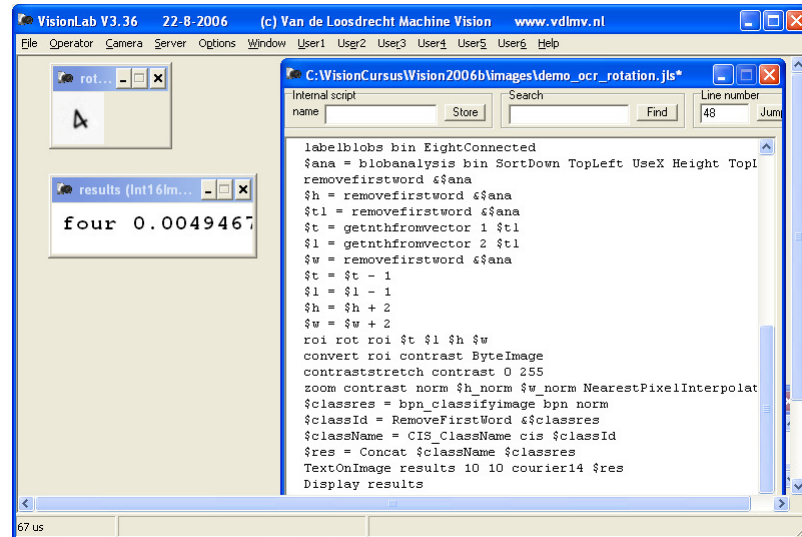


27-Aug-18

Classification with Neural
Networks

62

Demo with rotated input images

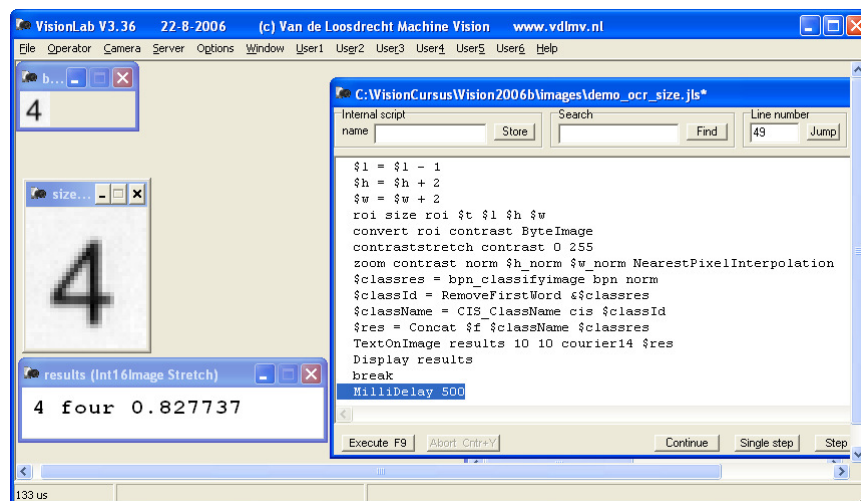


27-Aug-18

Classification with Neural
Networks

63

Demo with zoomed input images

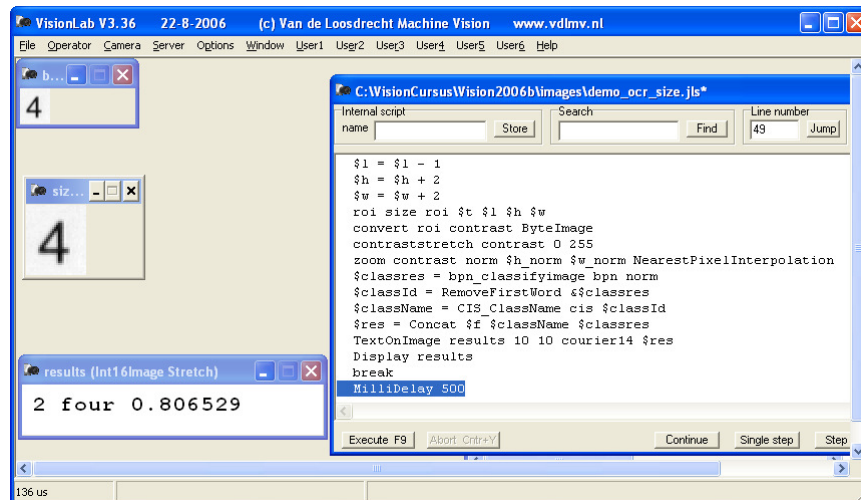


27-Aug-18

Classification with Neural
Networks

64

Demo with zoomed input images

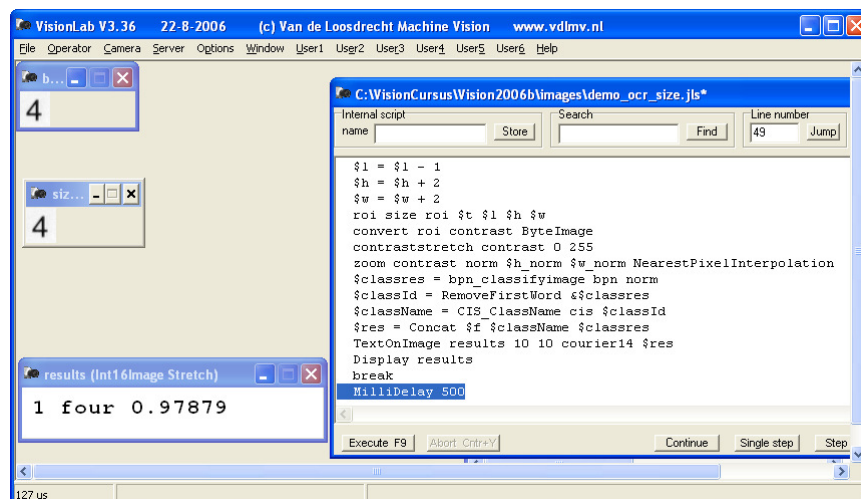


27-Aug-18

Classification with Neural
Networks

65

Demo with zoomed input images

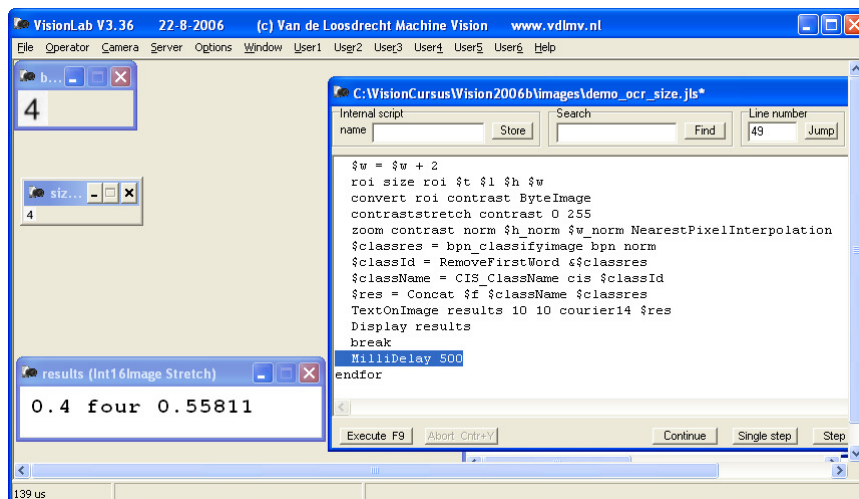


27-Aug-18

Classification with Neural
Networks

66

Demo with zoomed input images

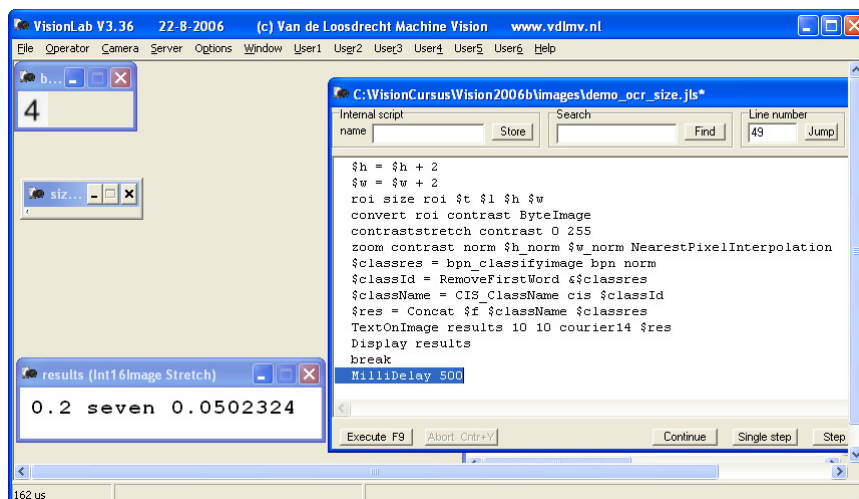


27-Aug-18

Classification with Neural
Networks

67

Demo with zoomed input images

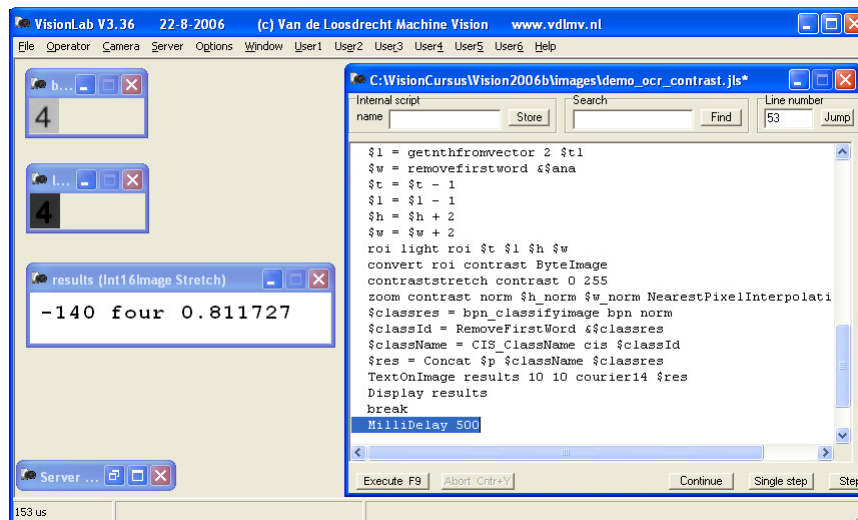


27-Aug-18

Classification with Neural
Networks

68

Demo with changed light conditions

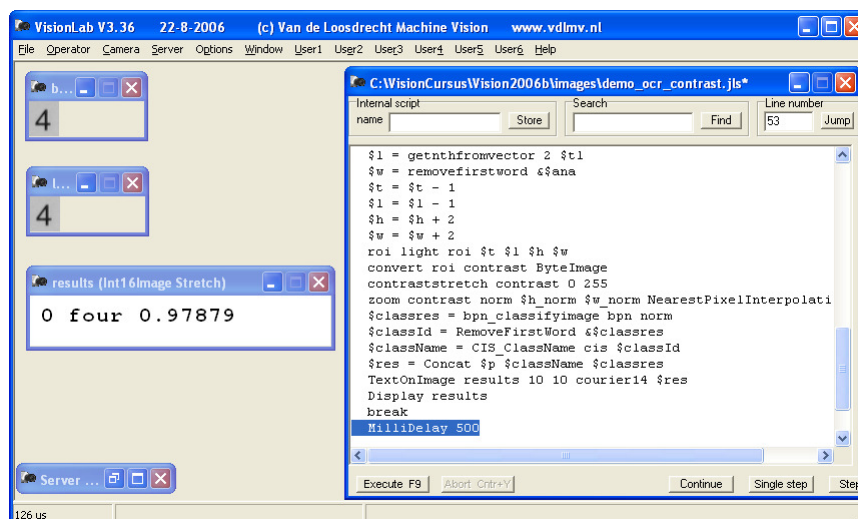


27-Aug-18

Classification with Neural
Networks

69

Demo with changed light conditions

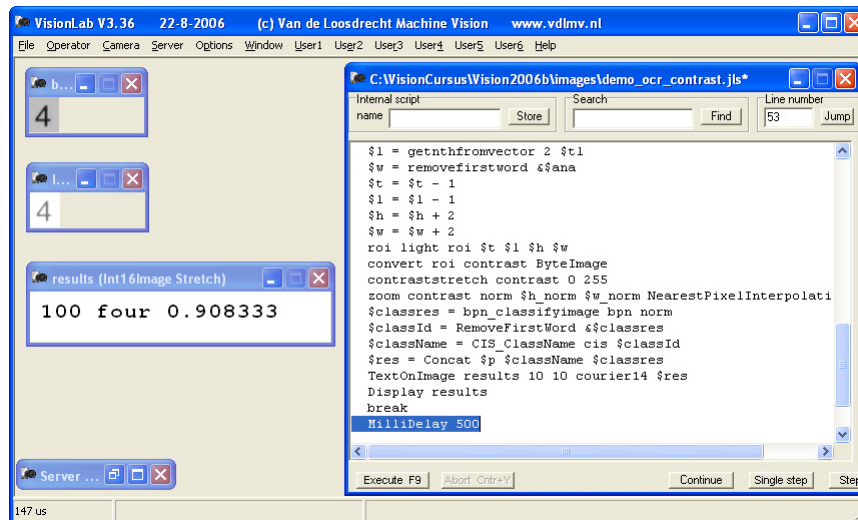


27-Aug-18

Classification with Neural
Networks

70

Demo with changed light conditions

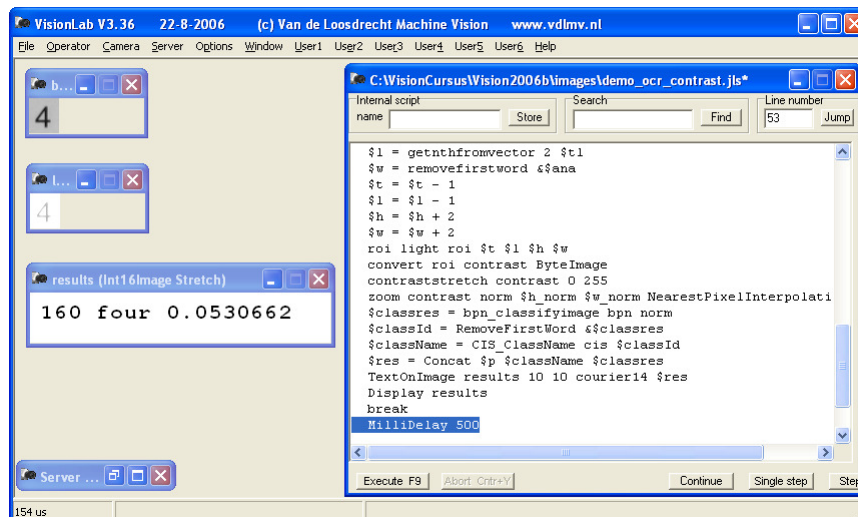


27-Aug-18

Classification with Neural
Networks

71

Demo with changed light conditions



27-Aug-18

Classification with Neural
Networks

72

Demo with wrong evaluation set (*)

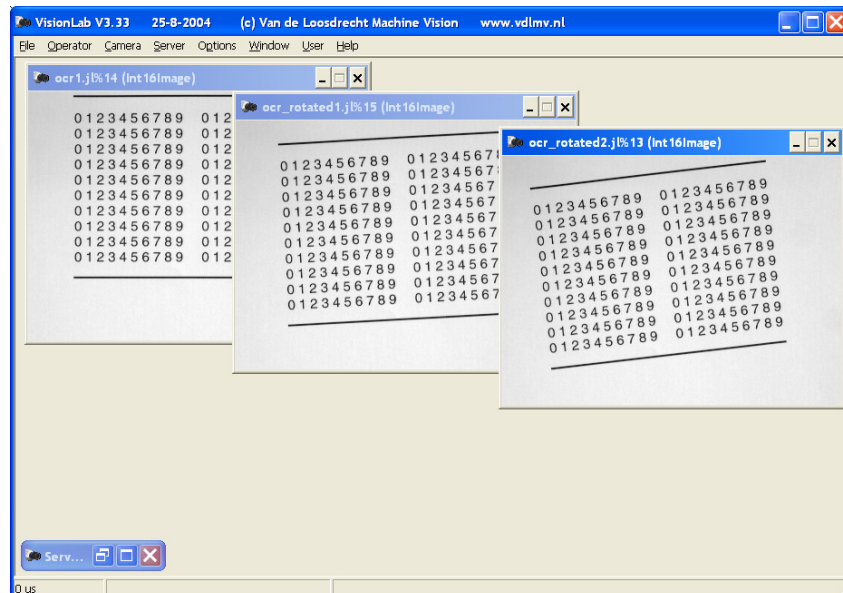
- The BPN was trained with numbers without rotation and is expected to perform worse if the digits are rotated
- Compare test images ocr1.jl, ocr_rotated1.jl and ocr_rotated2.jl
- For the rotated image ocr1 a CIS is generated with gen_ocr_rotated1_cis.jls
- For the rotated image ocr2 a CIS is generated with gen_ocr_rotated2_cis.jls
- Demonstrate training with ocr1.cis and evaluation sets the rotated CIS's, compare the results.
 - Analyse bad results for rotated2, have a look at the generated CIS !

27-Aug-18

Classification with Neural
Networks

73

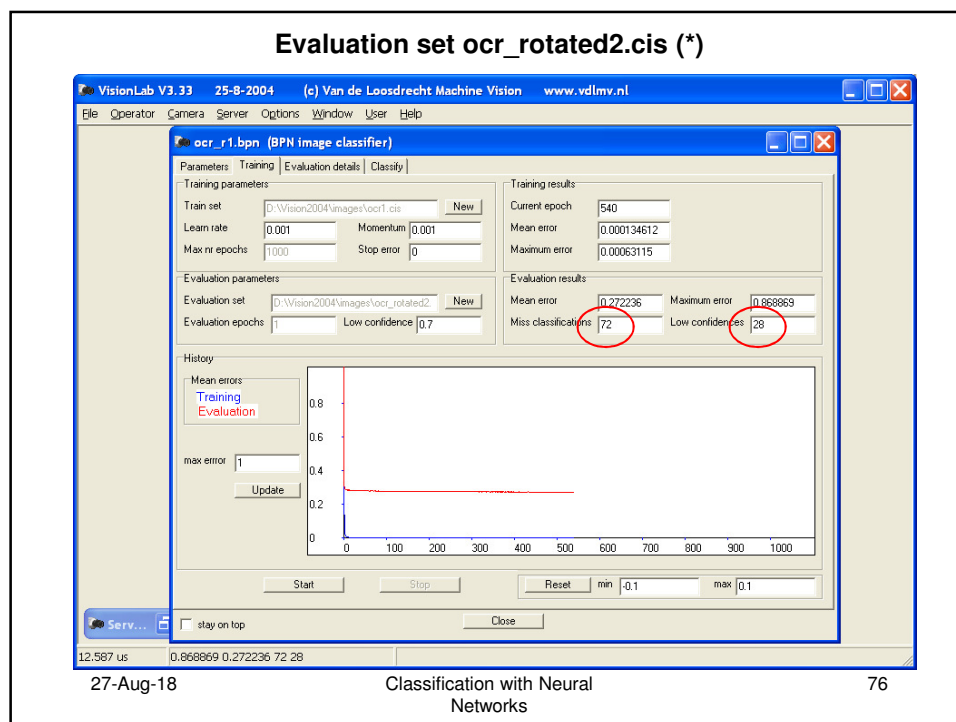
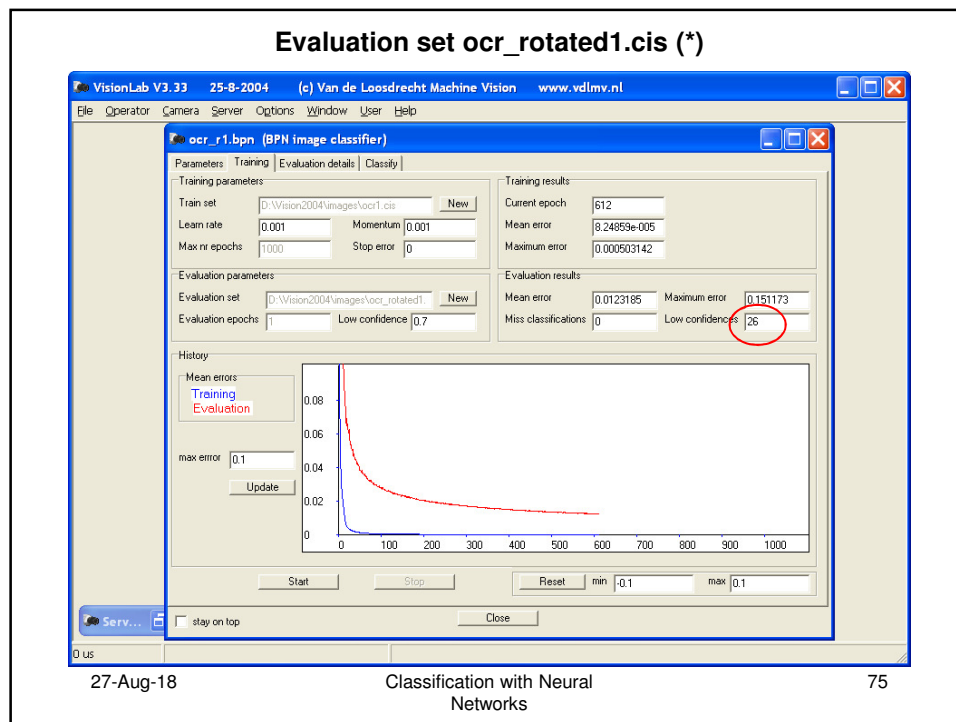
Comparing the rotated images (*)

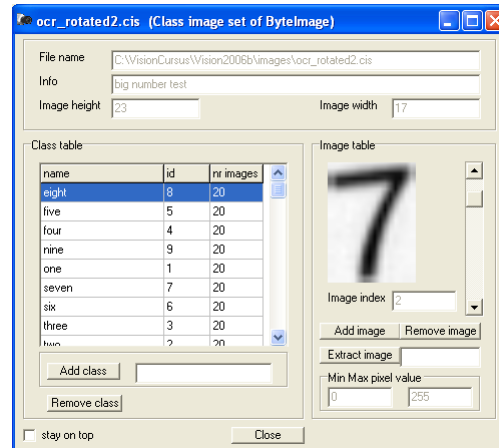


27-Aug-18

Classification with Neural
Networks

74



Wrong test data, seven in class “eight” ??? (*)

27-Aug-18

Classification with Neural
Networks

77

Testing with rotated input images (*)

- **Conclusion:**
 - The BPN can cope with small rotations, but the confidence level will drop for classifying
- **For applications:**
 - The training set can be extended with rotated images, but training will then be more difficult
 - All image are normalized for rotation before training or classification

27-Aug-18

Classification with Neural
Networks

78

Exercises rotation robustness OCR (*)

- **Normalize the training and evaluation sets for rotation and evaluate the BPN.**
Hints:
 - adapt script `gen_ocr_rotated2_cis.jls`
 - use the horizontal bars
- **Make new versions of image `ocr2.jl` and generate new CIS's for evaluation:**
 - **Scale, use zoom**
 - **Lighting, add pixel value**
- **Use new CIS's to evaluate the BPN**

- Answer first exercise: `gen_ocr_rotated_norm_cis.jls`,

27-Aug-18

Classification with Neural
Networks

79

Robustness OCR (*)

- **Image classification with BPN can be reasonable robust for**
 - **Scale**
 - **Rotation**
 - **Lighting**

This is because the images are normalized for these aspects

27-Aug-18

Classification with Neural
Networks

80

Feature vector classification (*)

- **Introduction**
- **Class Feature Sets (CFS)**
- **Training the BPN**
- **Using the BPN**
- **Exercise**

27-Aug-18

Classification with Neural
Networks

81

OCR feature vector classification (*)

- **Features are extracted from image and used for the classification**
- **Learning from features instead of pixels**
- **Data reduction**
- **OCR example:**
 - **Image: $23 \times 17 = 391$ pixels**
 - **Use features like:**
Area, AreaHoles, Breadth, CentreOfGravity,
MomentsScale_xy, NrofHoles, SumRows, SumCols, etc

27-Aug-18

Classification with Neural
Networks

82

Demo feature classifying (*)

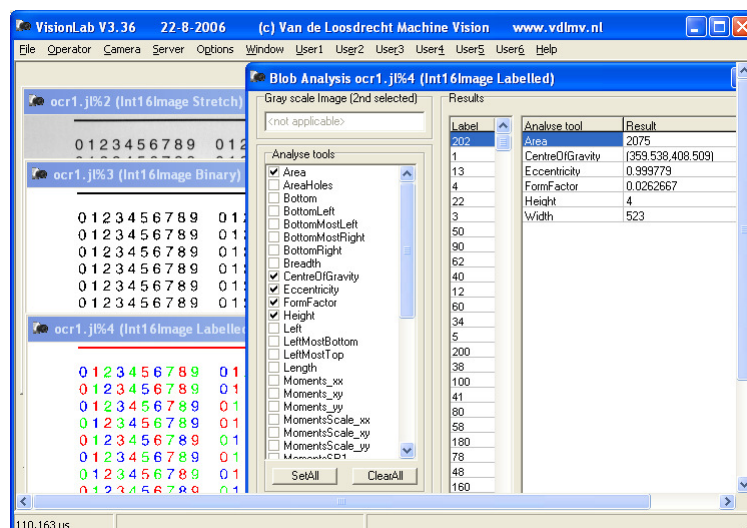
- { Demonstrate how to perform measurements
 - Open ocr1.jl (grayscale image)
 - Thresholdsodata (binair image)
 - LabelBlobs (labelled image)
 - BlobAnalyse (measurement)
- use test_feat_cfs.jls to generate test.cfs }
- Open test.cfs, show measurements
- CreateFeatureBPN with test.cfs, first hidden = 40 and nobias
- Demo learning, no feature: cg.x, cg.y and SumXXX
(use keyboard space to de-select and and activate selection with select button)
 - Learnrate = 0.002, momentum = 0.01
 - Problem with learning, about 20 miss classifications
 - Look at evaluation details:
 - Problems in classifying 6
 - Neuron output 6 and 9 = ~0.5, others ~0
- Use centre of gravity to distinguish between 6 and 9
see feature table in class feature set
 - 'six' : $7.3 < \text{cg.x} < 8.4$, $10.7 < \text{cg.y} < 12.1$
 - 'nine' : $8.0 < \text{cg.x} < 8.8$, $10.5 < \text{cg.y} < 11.6$

27-Aug-18

Classification with Neural
Networks

83

Perform measurements (*)

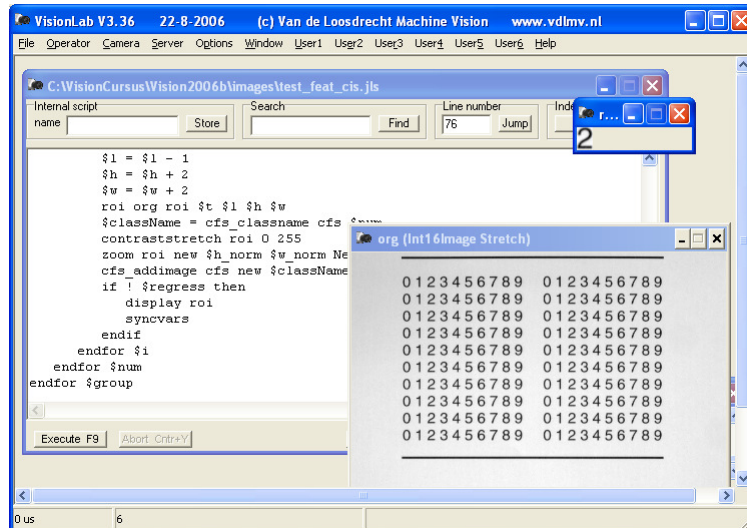


27-Aug-18

Classification with Neural
Networks

84

Generate Class Feature Set (*)



27-Aug-18

Classification with Neural Networks

85

Class Feature Set (CFS) (*)

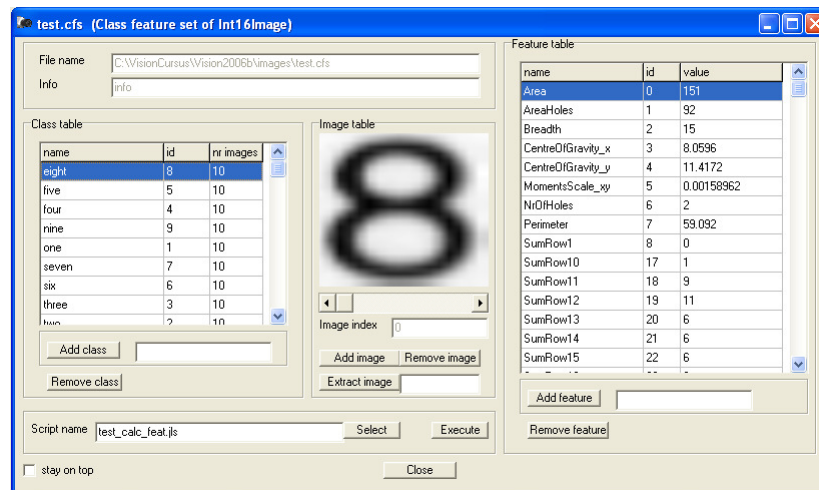
- **A CFS is a collection of images with their associated classes.**
All images in a CFS must have the same image type
- **CFS:**
 - **Class table with the name and id of each class**
 - **For each class an image table,**
each image in an image table has its unique image index number
 - **For each image a feature table,**
the features are the result of executing the script on the image

27-Aug-18

Classification with Neural Networks

86

Class Feature Set (*)

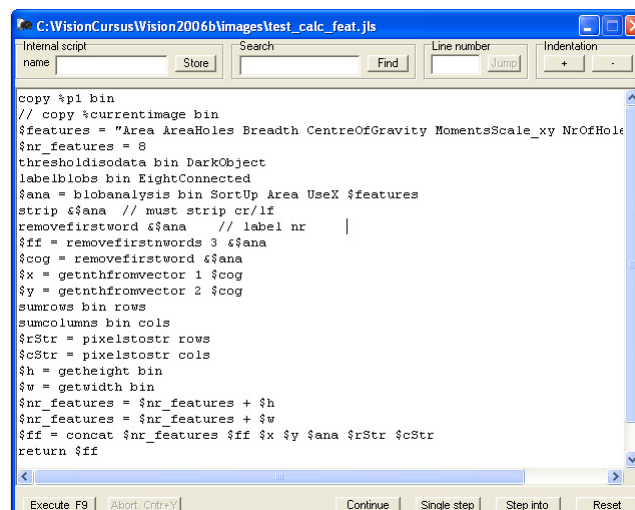


27-Aug-18

Classification with Neural
Networks

87

Script for calculating features (*)



27-Aug-18

Classification with Neural
Networks

88

Conventions for script used for CFS (*)

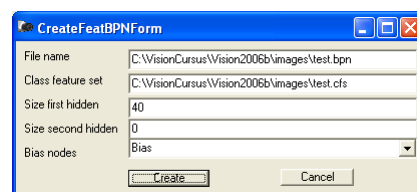
- Script is used for calculating the features for each image in the CFS
- First parameter (%p1) is name of image
- Function result is string with features separated by a space and in order of feature id

27-Aug-18

Classification with Neural
Networks

89

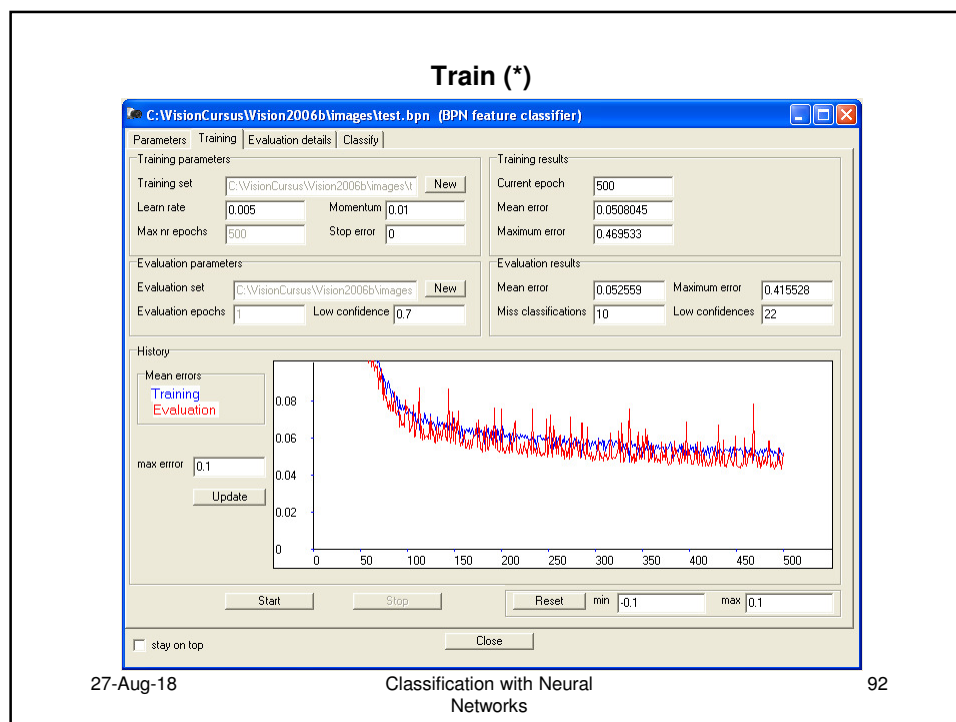
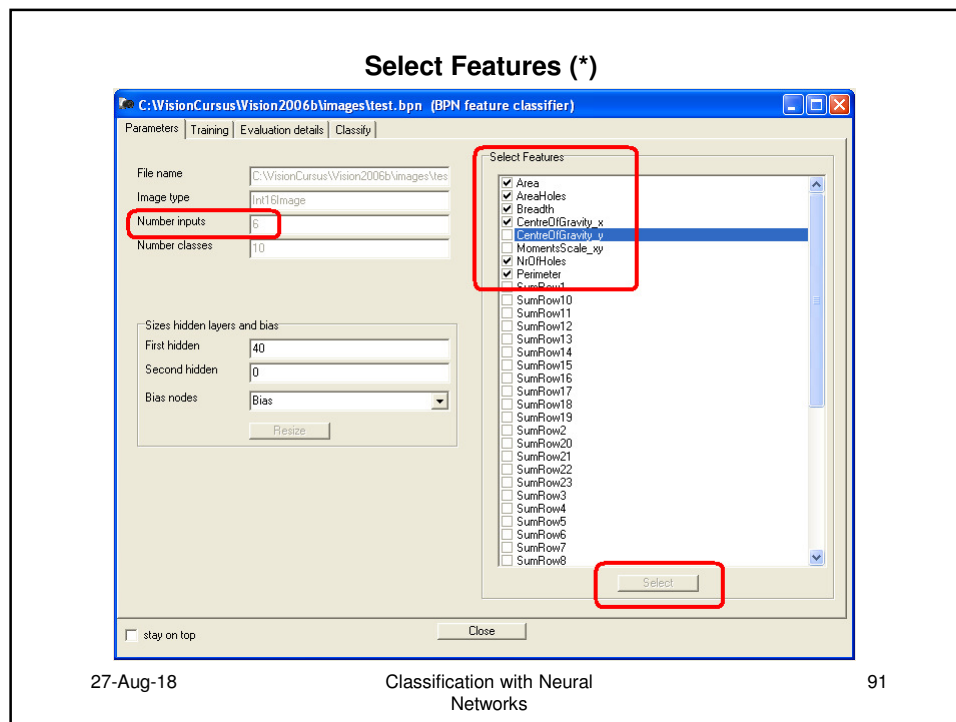
Create BPN Feature Classifier (*)



27-Aug-18

Classification with Neural
Networks

90



Problems in classifying 'six' and 'nine' (*)

C:\WisionCursus\Wision2006b\Images\test.bpn (BPN feature classifier)

Parameters	Training	Evaluation details	Classify			
expected	result	confidence	image index	zero	one	two
six	nine	0.292807	0	0.00185155	0.0765731	0.0671225
six	nine	0.464205	1	0.0250426	0.0279298	0.0615376
six	nine	0.294718	2	0.0234525	0.00580228	-0.0581282
six	nine	0.215449	3	0.00962974	-0.0127872	-0.0738007
six	nine	0.258903	4	0.00731135	0.000812119	-0.0796933
six	nine	0.192659	5	0.103674	0.0918519	0.0413702
six	nine	0.175205	6	0.0406201	0.00407151	-0.0611551
six	nine	0.333204	7	-0.000654991	0.0126087	-0.0630193
six	nine	0.0911839	8	0.0544959	0.004692	-0.076683
six	nine	0.239703	9	0.0040617	-0.0178151	-0.0724104
two	two	0.693265	5	0.0292318	0.054167	0.976915
three	three	0.674879	1	-0.00571292	0.0143831	0.0818192
five	five	0.584216	0	-0.00703796	-0.00363368	-0.107804
five	five	0.631393	2	-0.0123274	-0.0400246	0.00346006
seven	seven	0.648074	8	0.00616532	0.0221877	0.0449123
eight	eight	0.554605	1	0.050642	0.0315463	-0.112407
eight	eight	0.687671	2	-0.0237613	0.0192384	-0.0676298
eight	eight	0.653727	3	0.000991318	-0.00657891	-0.00270528
eight	eight	0.688809	4	-0.00677057	-0.00480033	-0.044378
two	two	0.633265	5	0.0292318	0.054167	0.976915
three	three	0.674879	1	-0.00571292	0.0143831	0.0818192
five	five	0.584216	0	-0.00703796	-0.00363368	-0.107804
five	five	0.631393	2	-0.0123274	-0.0400246	0.00346006
seven	seven	0.648074	8	0.00616532	0.0221877	0.0449123
eight	eight	0.554605	1	0.050642	0.0315463	-0.112407
eight	eight	0.687671	2	-0.0237613	0.0192384	-0.0676298
eight	eight	0.653727	3	0.000991318	-0.00657891	-0.00270528
eight	eight	0.688809	4	-0.00677057	-0.00480033	-0.044378

HighDetailsChange detailsExtract Imagemiss

☐ stay on topClose

27-Aug-18

Classification with Neural
Networks

93

Classifier hesitates between 'six' and 'nine' (*)

C:\WisionCursus\Wision2006b\Images\test.bpn (BPN feature classifier)

Parameters	Training	Evaluation details	Classify			
three	four	five	six	seven	eight	nine
-0.0863201	-0.00773896	0.0561821	0.386059	0.00702394	0.0368196	0.705747
-0.014489	-0.0163096	-0.00386354	0.251171	0.0144926	0.0323151	0.686393
-0.0193762	0.0155715	-0.0093291	0.357905	0.023955	-0.0341207	0.657022
0.0409866	0.0453189	-0.0381982	0.386121	0.0195316	-0.0334699	0.598189
-0.0128924	0.0270091	-0.0166521	0.38786	0.0260063	-0.0463914	0.65886
0.133183	-0.00986574	-0.00811043	0.425509	0.085187	0.0608606	0.628591
0.0739654	0.0175539	-0.0446945	0.410464	0.00577235	-0.01815	0.584848
-0.049207	0.0030565	0.00673095	0.35788	0.0272364	-0.0409723	0.715552
0.11894	0.0247009	-0.0581103	0.456634	0.0217727	-0.0202445	0.548264
0.026185	0.0584498	-0.0320488	0.307712	0.032544	-0.036118	0.599572
0.176968	0.0422569	0.0184501	-0.0120175	0.0889236	0.0276098	0.0685542
0.814069	-0.0272905	0.158024	-0.0467461	-0.069494	0.00744601	-0.0263376
0.237666	0.0209242	0.905552	-0.174813	-0.0362203	-0.0186547	0.196389
-0.145275	0.038063	1.09845	-0.216747	-0.00523965	-0.0318645	0.24891
0.000483778	0.0346999	0.10198	0.232265	1.0877	0.069519	-0.162916
0.109946	0.0540324	-0.0884168	-0.122714	0.0456769	0.95408	0.273344
0.00770729	-0.0193156	-0.0378557	-0.153926	-0.0144761	0.994972	0.184134
-0.0509626	-0.00473029	-0.0177529	-0.161339	0.0017225	0.983996	0.206039
-0.0397874	-0.0211938	-0.033239	-0.162405	-0.0115563	0.978845	0.180368

HighDetailsChange detailsExtract Imagemiss

☐ stay on topClose

27-Aug-18

Classification with Neural
Networks

94

Add extra features (*)

- **Analyse evaluation details:**
 - Problems in classifying 'six'
 - Neuron output 'six' and 'nine' ~ 0.5, others ~0
- Use centre of gravity to distinguish between 6 and 9
see feature table in class feature set
 - 'six' : $7.3 < \text{cg.x} < 8.4$, $10.7 < \text{cg.y} < 12.1$
 - 'nine' : $8.0 < \text{cg.x} < 8.8$, $10.5 < \text{cg.y} < 11.6$
- cg.x does NOT discriminate between 6 and 9,
there is overlap between 8.0 and 8.4
testing shows the cg depends on threshold
number (and its cg) moves up and down

27-Aug-18

Classification with Neural
Networks

95

Features for 'six' (*)

test.cfs (Class feature set of Int16Image)

File name: C:\VisionCourse\Vision2006b\Images\test.cfs
Info: info

Class table

name	id	nr images
four	4	10
nine	9	10
one	1	10
seven	7	10
six	6	10
three	3	10
two	2	10
zero	0	10

Image table

Image index: 0

Feature table

name	id	value
Area	0	136
AreaHoles	1	61
Breadth	2	15.0357
CentreOfGravity_x	3	8
CentreOfGravity_y	4	10.7721
MomentsScale_xy	5	0.0078936
NrOfHoles	6	1
Perimeter	7	77.973
SumRow1	8	0
SumRow10	17	7
SumRow11	18	10
SumRow12	19	7
SumRow13	20	6
SumRow14	21	4
SumRow15	22	3

Script name: test_calc_feats.js

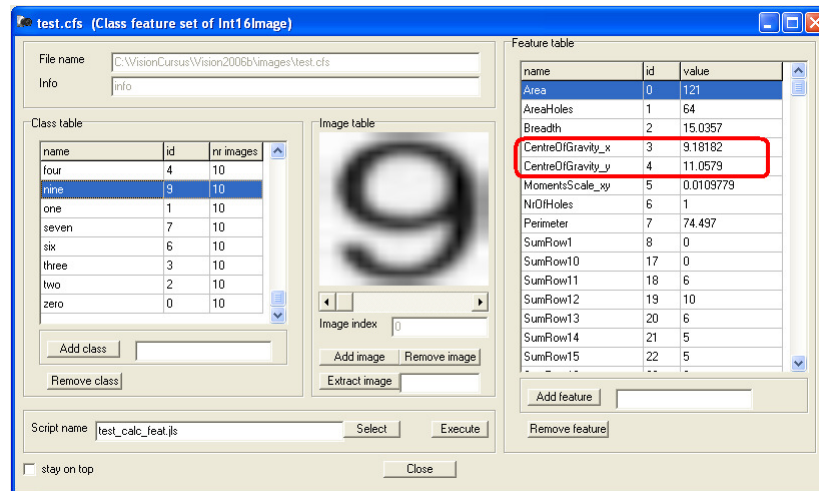
Buttons: Add class, Remove class, Add image, Remove image, Extract image, Add feature, Remove feature, Execute, Close, stay on top

27-Aug-18

Classification with Neural
Networks

96

Features for 'nine' (*)

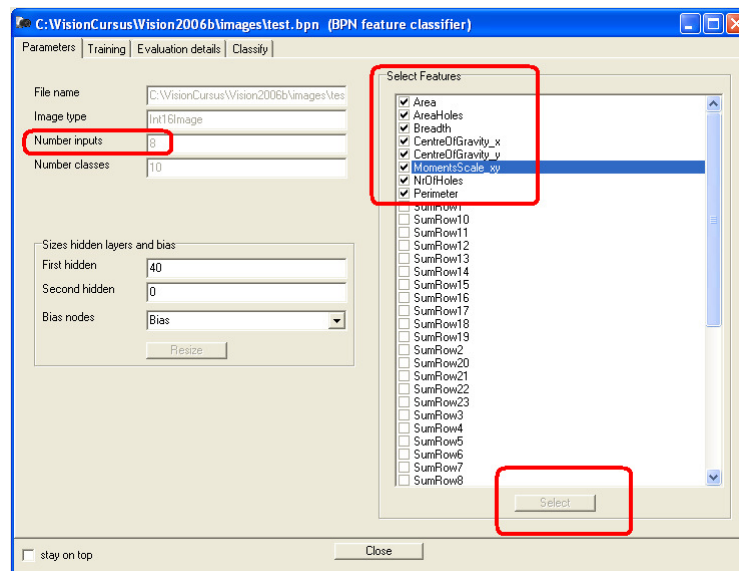


27-Aug-18

Classification with Neural
Networks

97

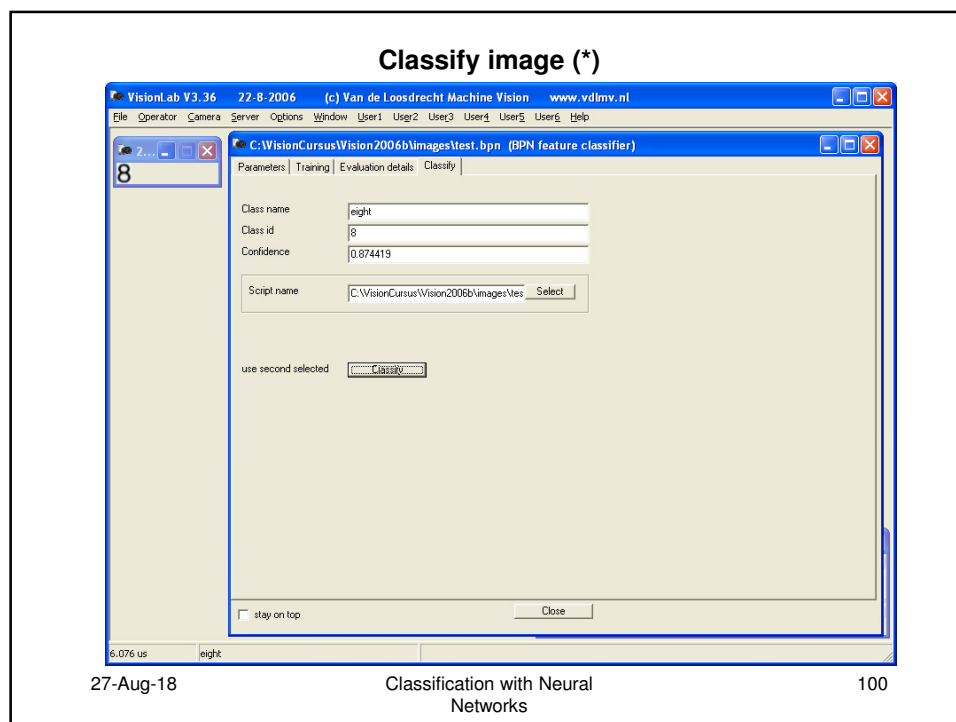
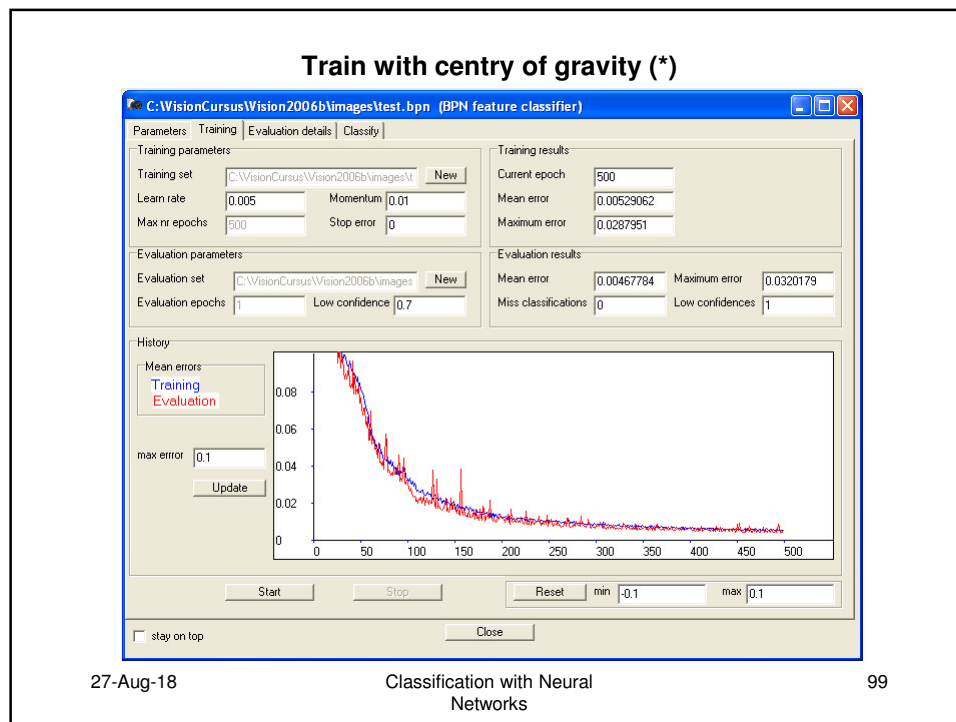
Select extra features cg.x and cg.y (*)



27-Aug-18

Classification with Neural
Networks

98



OCR feature vector classification (*)

Conclusions:

- Complexity has been reduced from 391 dimensions to 8
- Selection of which features to use is can be quite difficult

Note:

Invariant to scale, rotation and lighting can be implemented:

- In feature extraction
- By normalizing the images

27-Aug-18

Classification with Neural
Networks

101

Optimizing the settings for a feature BPN (*)

This function uses a Genetical Algorithm (GA) in order to find the best solution.

NOTE: this optimization process is very time consuming and for real problems it may take hours or even days.

In the Low/High box the user must specify the search space for size of hidden layers, learn rate and momentum, by specifying the lowest and highest border values.

In the settings box must be specified:

- the population size for the GA algorithm
- the number of epochs the BPN is trained before being evaluated
- the low confidency factor used in the evaluation
- the maximum number of generation for the GA algorithm

27-Aug-18

Classification with Neural
Networks

102

Optimizing the settings for a feature BPN (*)

In the detailed setting box, specialized setting can be set, it is advised to use the standard values.

- the MinError is the minimal target error
- the DeltaError is the minimal error for start of hill climbing
- the MicroP is the probability during self mate that parameter is a micro mutation

For feature classifiers the features to select from box can be used to select the features that take part in the optimizing process.

With the Start button the optimising process is started. With the Stop button the optimising process is stopped, but first the current generation is optimised completely, note this can take some time!

27-Aug-18

Classification with Neural
Networks

103

Optimizing the settings for a feature BPN (*)

In the best result box the best values found sofar for hidden layers, learn rate and momentum are displayed. Also the current generation number and the best error value found.

For feature classifiers the best features box shows the best combination of features found.

With the Accept button the best solution found is used to initialise the settings in the Parameters and Training tab. Then the normal full training is started.

27-Aug-18

Classification with Neural
Networks

104

Demonstration genetic algorithm for optimizing feature BPN (*)

- Go to GA Optimize form, this can be used to find the optimal setting for the BPN
Use settings:
Low and High first hidden = 40 (is fixed for the demo)
Low and High second hidden = 0
LearnRate: 0.0001 – 0.01
Momentum: 0 – 0.01
Populationsize: 4
NrEpochs: 10
LowConfidence: 0.5
MinError: 0
DeltaError: 1
MicroP: 0.25
- Select first 8 features
Click start to start optimise process
Note this process is very time consuming, for the demo PopulationSize and NrEpochs have been set to low values.
- Click stop button to interrupt the optimise process
- Select first all SumXX features
Click start to start optimise process
- Click accept button to initialise the settings in the Parameters and Training tab. Then the normal full training is started.

27-Aug-18

Classification with Neural
Networks

105

Set the parameters for the optimising process (*)

The screenshot shows the 'test_num.bpn (BPN feature classifier)' window with the 'GA Optimize' tab selected. The window contains several sections for configuring the genetic algorithm:

- Low/High settings:** A table with columns 'Low' and 'High' for 'First hidden' (40, 40), 'Second hidden' (0, 0), 'Learn rate' (0.0001, 0.01), and 'Momentum' (0, 0.01).
- Settings:** Fields for 'Population size' (4), 'Nr Epochs' (10), 'Low confidence' (0.5), and 'Max generations' (100).
- Features to select from:** A list box containing 'Area', 'AreaHoles', 'Breadth', 'CentreOfGravity_x', 'CentreOfGravity_y', 'MomentsScale_xy', 'NrOfHoles', 'Perimeter', 'SumRow1', and 'SumRow2'. 'Perimeter' is currently selected.
- Best results:** Fields for 'Generation Nr' (43) and 'Error' (0.480168).
- Detailed settings:** Fields for 'Min Error' (0), 'Delta Error' (1), and 'MicroP' (0.25).
- Best features:** A list box containing 'Area', 'AreaHoles', 'Breadth', 'CentreOfGravity_x', 'CentreOfGravity_y', 'MomentsScale_xy', and 'Perimeter'.

At the bottom, there are buttons for 'Start', 'Stop', 'Accept', and 'Close'. A checkbox 'stay on top' is also present.

27-Aug-18

Classification with Neural
Networks

106

Set the parameters for the optimising process (*)

test_num.bpn (BPN feature classifier)

Parameters | Training | Evaluation details | Classify | GA Optimize

Low/High

	Low	High
First hidden	40	40
Second hidden	0	0
Learn rate	0.0001	0.01
Momentum	0	0.01

Best results

Generation Nr: 11
Error: 0.174863

Settings

Population size: 4
Nr Epochs: 10
Low confidence: 0.5
Max generations: 100

Detailed settings

Min Error: 0
Delta Error: 1
MicroP: 0.25

Features to select from

- ☐ Area
- ☐ AreaHoles
- ☐ Breadth
- ☐ CentreOfGravity_x
- ☐ CentreOfGravity_y
- ☐ MomentsScale_xy
- ☐ NoOfHoles
- ☒ Parameter
- ☒ SumRow1
- ☒ SumRow2

Mark all | Unmark all

Start | Stop | Accept

☐ stay on top | Close

27-Aug-18

Classification with Neural
Networks

107

Used find optimal parameters to start the full training (*)

test_num.bpn (BPN feature classifier)

Parameters | Training | Evaluation details | Classify | GA Optimize

Training parameters

Training set: D:\VisionCursus\Vision2009e\images\ | New

Learn rate: 0.0084706 | Momentum: 0.0051927

Max nr epochs: 1000 | Stop error: 0

Evaluation parameters

Evaluation set: D:\VisionCursus\Vision2009e\images\ | New

Evaluation epochs: 1 | Low confidence: 0.6

Training results

Current epoch: 1000
Mean error: 0.00119274
Maximum error: 0.00406956

Evaluation results

Mean error: 0.00107894 | Maximum error: 0.00393106
Miss classifications: 0 | Low confidences: 0

History

Mean errors

Training (blue line)
Evaluation (red line)

max error: 0.1 | Update

Start | Stop | Reset | min: -0.1 | max: 0.1

☐ stay on top | Close

27-Aug-18

Classification with Neural
Networks

108

Exercise feature vector classification (*)

Experiment with adding and removing features

Tip: experiment with SumRows and SumCols only

27-Aug-18

Classification with Neural
Networks

109

OCR feature vector classification with scale, size and position invariant features (*)

Select from the BlobAnlyse tool the measurements which calculate scale, size and position independent features:

- Eccentricity
- FormFacor
- Hu1
- Hu2
- Hu3
- Hu4
- Hu5
- Hu6
- Hu7
- NrOfHoles

27-Aug-18

Classification with Neural
Networks

110

OCR feature vector classification with scale, size and position invariant features (*)

Demonstration:

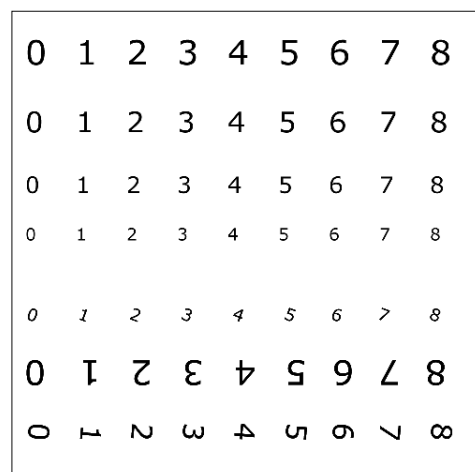
- Create train set
- Train Feature Classifier
- Evaluate Feature Classifier

27-Aug-18

Classification with Neural
Networks

111

Use image hu_test_train.jl as train set (*)



27-Aug-18

Classification with Neural
Networks

112

Script for calculation the features (*)

```

copy %pl_bin
$$nr_features = 10
ThresholdIsodata _bin DarkObject
FillSpecificHoles _bin FourConnected Area 1 8 UseX
labelblobs _bin EightConnected
$$ana = blobanalysis _bin NoSort Area UseX Eccentricity FormFactor Hu1 Hu2
strip $$ana // must strip cr/lf
removefirstword $$ana // label nr
$$ff = concat $$nr_features $$ana
return $$ff

```

27-Aug-18

Classification with Neural
Networks

113

Use script gen_hu_num_train_cfs.jls to generate train CFS (*)

name	id	nr images
zero	0	7
one	1	7
two	2	7
three	3	7
four	4	7
five	5	7
six	6	7
seven	7	7
eight	8	7

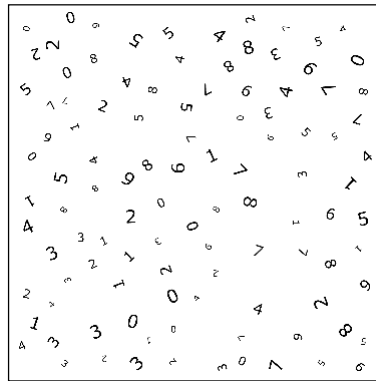
name	id	value
Eccentricity	0	0.0937896
FormFactor	1	0.523789
Hu1	2	-0.345225
Hu2	3	-1.7183
Hu3	4	-5.61924
Hu4	5	-5.97337
Hu5	6	-11.948
Hu6	7	7.26741
Hu7	8	-11.8955
NoOfHoles	9	1

27-Aug-18

Classification with Neural
Networks

114

Use image hu_test_eval.jl as evaluation set (*)

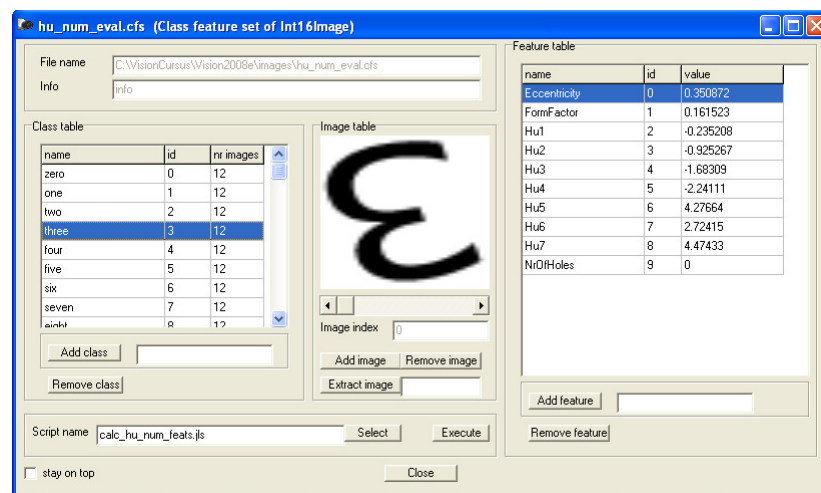


27-Aug-18

Classification with Neural Networks

115

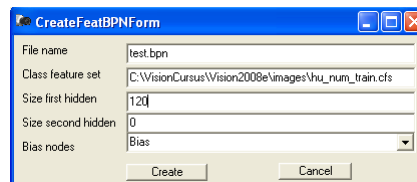
Evaluation set hu_num_eval.cfs (*)



27-Aug-18

Classification with Neural Networks

116

Create BPN Feature Classifier (*)

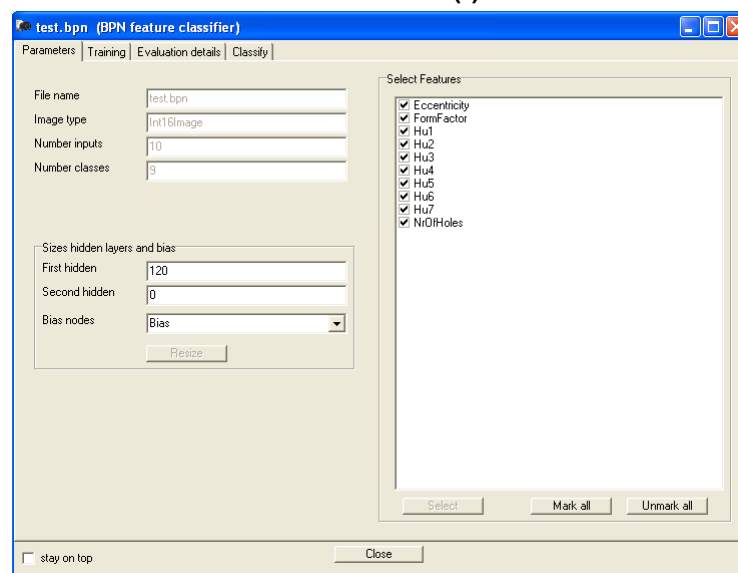
A dialog box titled "CreateFeatBPNForm" with a blue title bar. It contains the following fields and controls:

- File name: text box containing "test.bpn"
- Class feature set: text box containing "C:\VisionCursus\Vision2008e\Images\hu_num_train.cfs"
- Size first hidden: text box containing "120"
- Size second hidden: text box containing "0"
- Bias nodes: dropdown menu showing "Bias"
- Buttons: "Create" and "Cancel"

27-Aug-18

Classification with Neural
Networks

117

Select Features (*)

A dialog box titled "test.bpn (BPN feature classifier)" with a blue title bar. It has four tabs: "Parameters", "Training", "Evaluation details", and "Classify". The "Parameters" tab is active.

Parameters section:

- File name: text box containing "test.bpn"
- Image type: text box containing "int16Image"
- Number inputs: text box containing "10"
- Number classes: text box containing "3"

Sizes hidden layers and bias section:

- First hidden: text box containing "120"
- Second hidden: text box containing "0"
- Bias nodes: dropdown menu showing "Bias"
- Button: "Resize"

Select Features section:

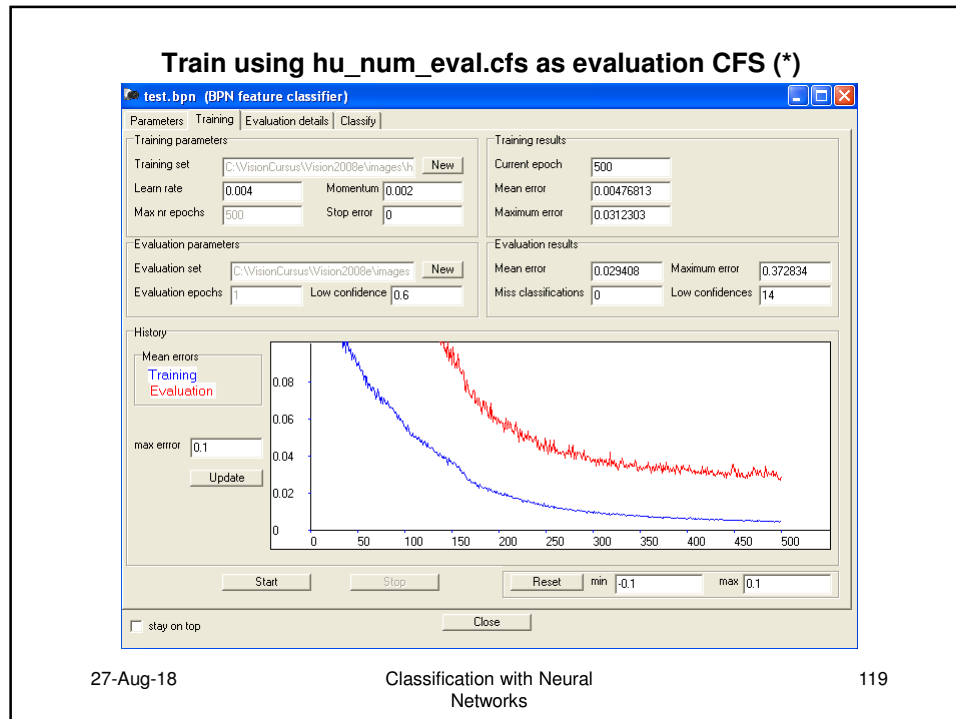
- Checkboxes (all checked): Eccentricity, FormFactor, Hu1, Hu2, Hu3, Hu4, Hu5, Hu6, Hu7, NrOfHoles
- Buttons: "Select", "Mark all", "Unmark all"

Footer: "stay on top" checkbox, "Close" button

27-Aug-18

Classification with Neural
Networks

118



Exercise feature vector classification (*)

Experiment with adding and removing features and tuning parameters neural network

Switch training and evaluation set and restart training, explain why result is much better