



Computer Vision

Classification with Neural Networks

27 August 2008

Copyright © 2001 – 2008 by
Noordelijke Hogeschool Leeuwarden and Van de Loosdrecht Machine Vision
All rights reserved

j.van.de.loosdrecht@tech.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

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

28-Aug-08

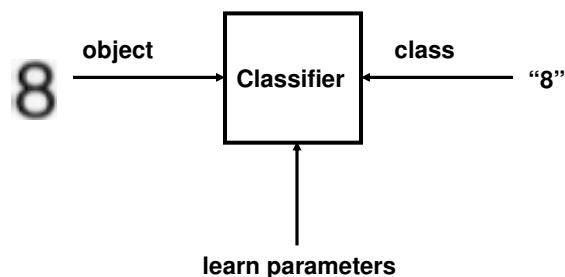
Classification with Neural Networks

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



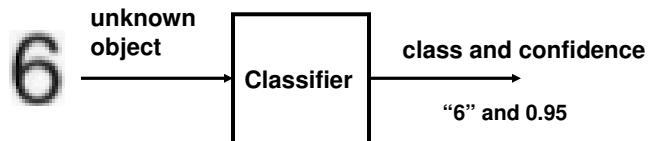
28-Aug-08

Classification with Neural Networks

4

Introduction classification

- Production mode



28-Aug-08

Classification with Neural Networks

5

Introduction classification

- Image classification
 - ‘raw’ image data
- Feature vector classification
 - features are extracted from image and used for the classification
 - data reduction

28-Aug-08

Classification with Neural Networks

6

Introduction neural networks

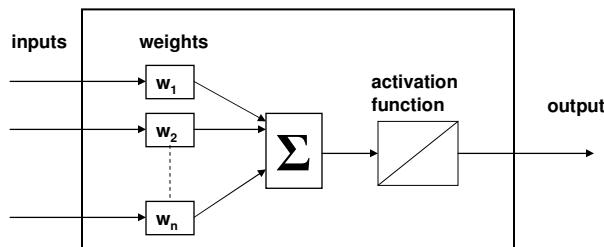
- Based on a model of our brain
- Neuron
- Neural networks
- Back Propagation Network
- Example: learning sinus function

28-Aug-08

Classification with Neural Networks

7

Neuron



28-Aug-08

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 Back Propagation Network (BPN)

28-Aug-08

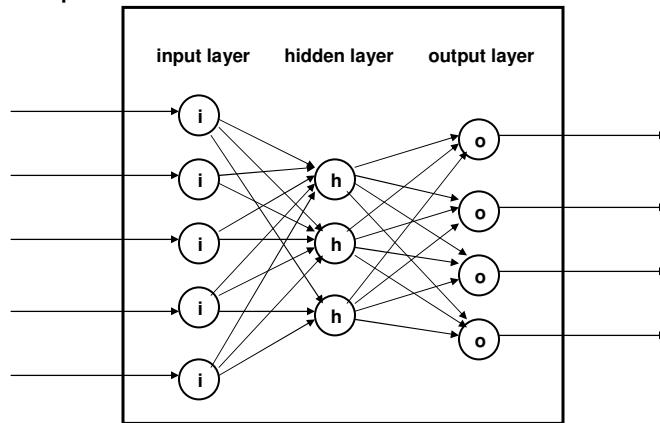
Classification with Neural Networks

9

Back Propagation Network

Example with:

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



28-Aug-08

Classification with Neural Networks

10

Example: learning sinus function

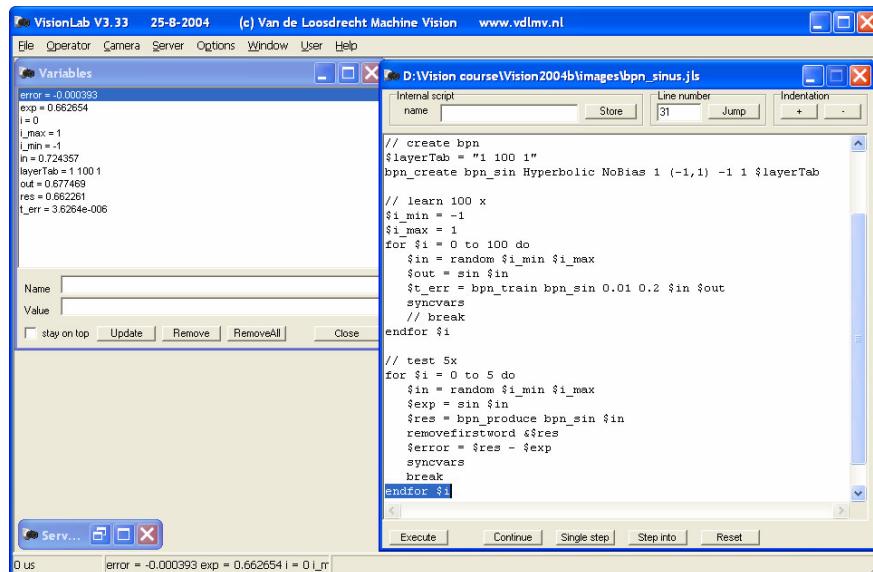
- Create BPN with
 - Input layer of 1 neuron
 - One hidden layer of 100 neurons
 - Output layer of 1 neuron
- Train the BPN with 100 random examples: $x, \sin(x)$
 - $\$in = \text{random } -1 1$
 - $\$out = \sin \in
 - Note the behaviour of the training error ($\$t_err$)
- Test the BPN with 5 random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

28-Aug-08

Classification with Neural Networks

11

Example: learning sinus function



The screenshot shows the VisionLab V3.33 software interface. The main window displays a script titled "D:\Vision course\Vision2004b\images\bpm_sinus.jls". The script content is as follows:

```

// create bpm
$layerTab = "1 100 1"
bpm_create bpm_sin Hyperbolic NoBias 1 (-1,1) -1 1 $layerTab

// learn 100 x
$i_min = -1
$i_max = 1
for $i = 0 to 100 do
  $in = random $i_min $i_max
  $out = sin $in
  $t_err = bpm_train bpm_sin 0.01 0.2 $in $out
  syncvars
  // break
endfor $i

// test 5x
for $i = 0 to 5 do
  $in = random $i_min $i_max
  $exp = sin $in
  $res = bpm_produce bpm_sin $in
  removefirstword $res
  $error = $res - $exp
  syncvars
  break
endfor $i

```

The "Variables" window on the left shows the following variable values:

Name	Value
error	-0.000393
exp	0.662654
i	0
i_max	1
in	0.724357
layerTab	1 100 1
out	0.677469
res	0.662261
t_err	3.6264e-006

28-Aug-08

Classification with Neural Networks

12

Image classification

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

28-Aug-08

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

28-Aug-08

Classification with Neural Networks

14

Class Image Set (CIS)

- A CIS is a collection of images with their associated classes.
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

28-Aug-08

Classification with Neural
Networks

15

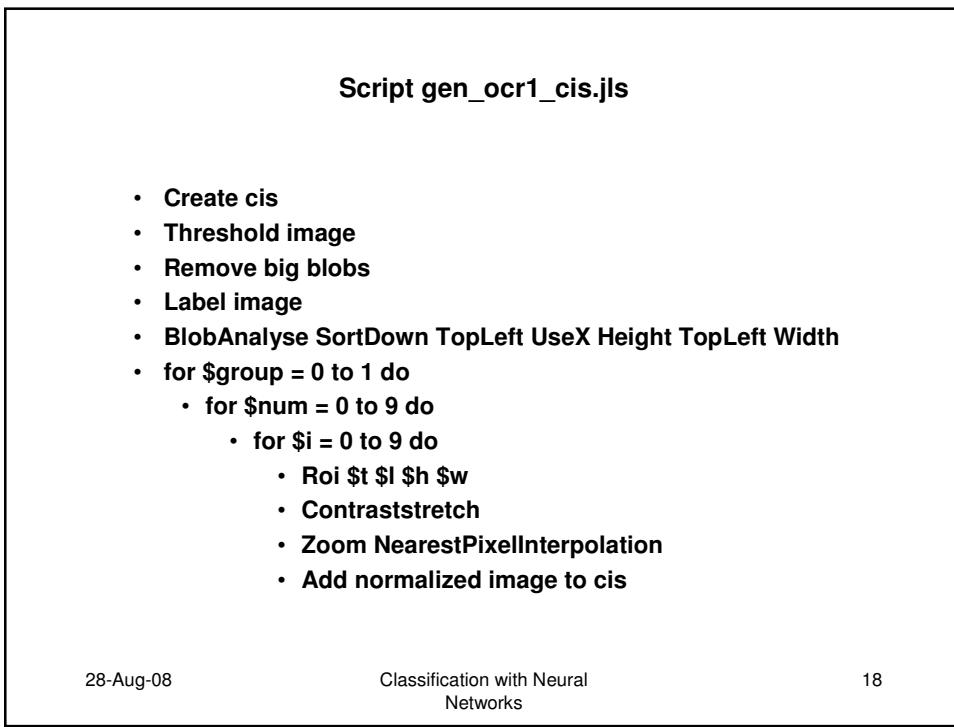
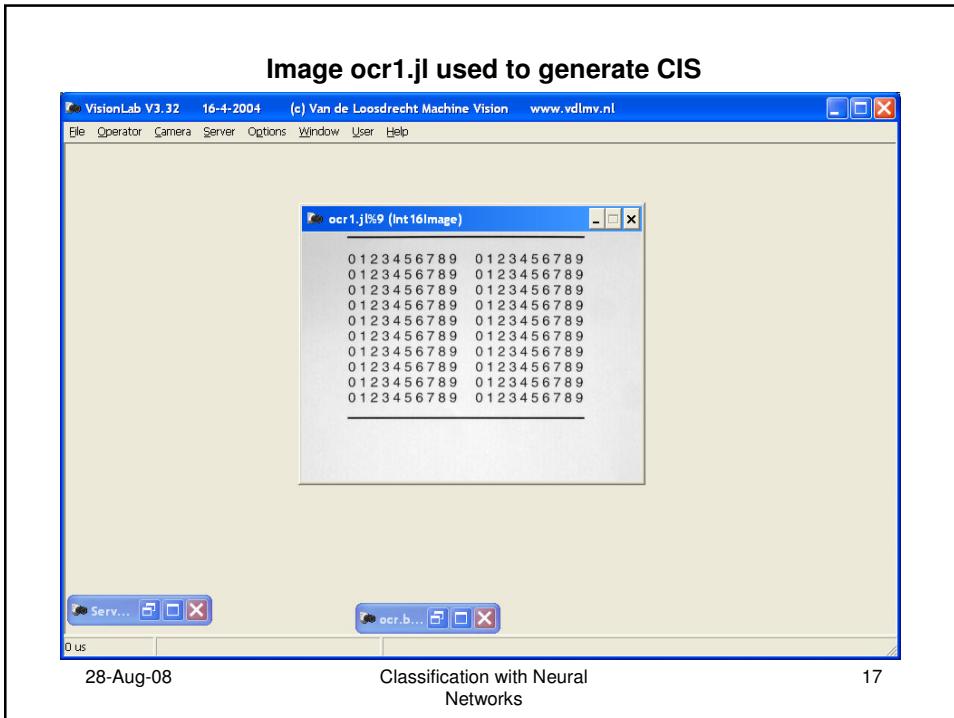
Demonstration Class Image Set (CIS) (*)

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

28-Aug-08

Classification with Neural
Networks

16



Demonstration Class Image Set (CIS)

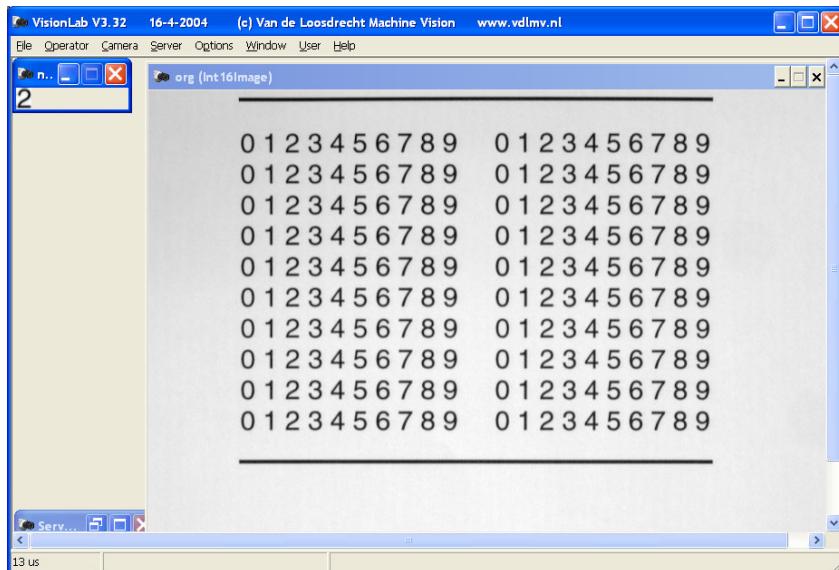
- Use script gen_ocr1_cis.jls to generate ocr1.cis,
- Open and examine ocr1.cis

28-Aug-08

Classification with Neural Networks

19

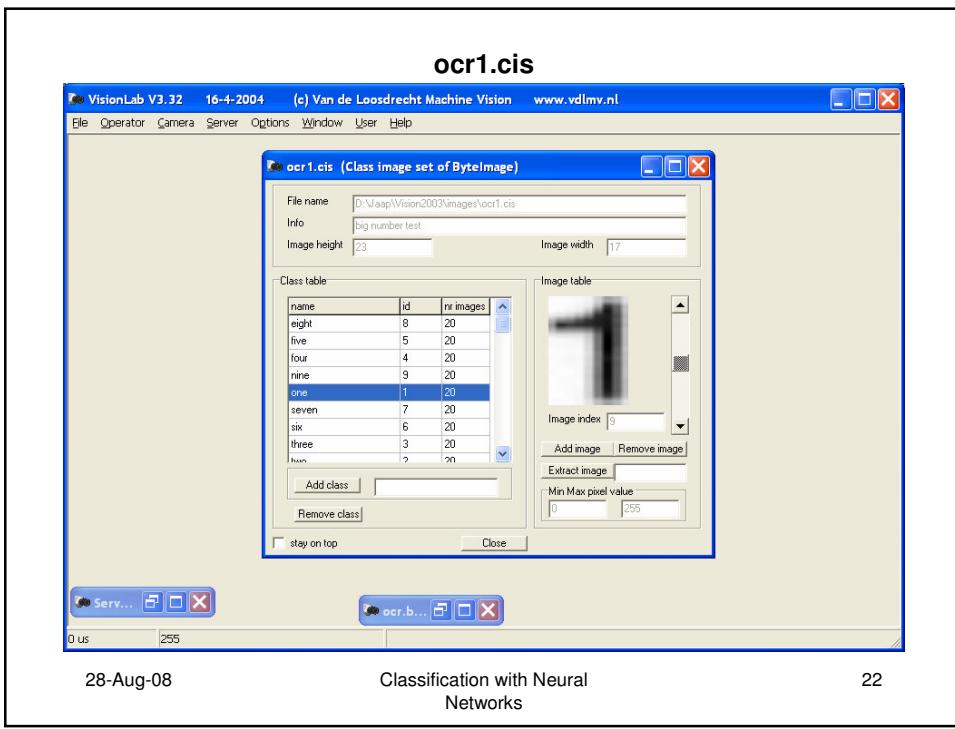
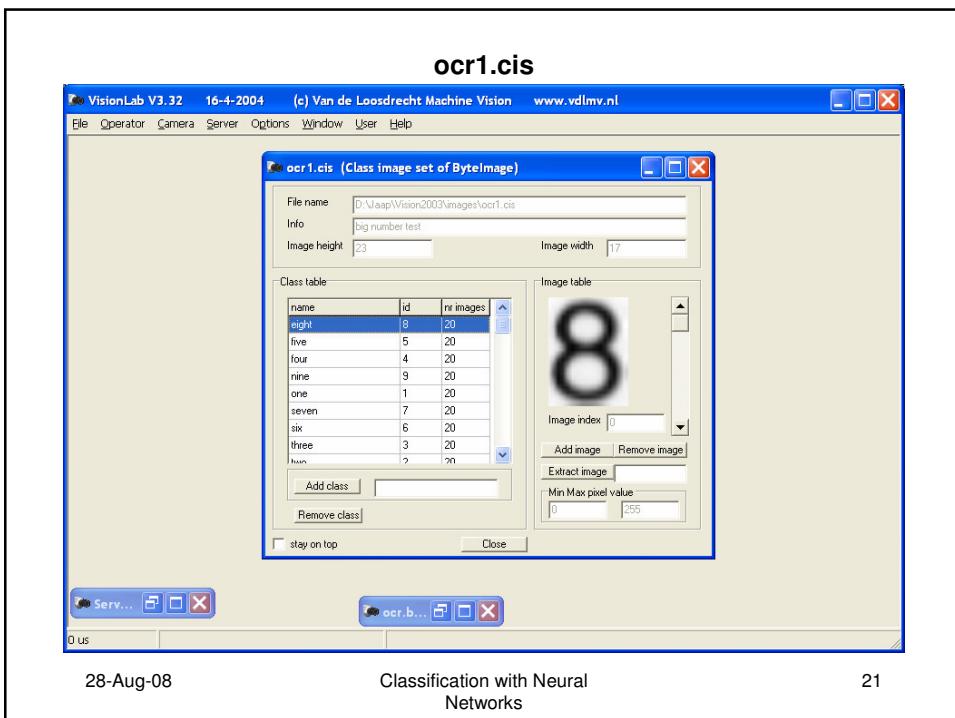
Generation of ocr.cis



28-Aug-08

Classification with Neural Networks

20



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

28-Aug-08

Classification with Neural Networks

23

Training the BPN

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

28-Aug-08

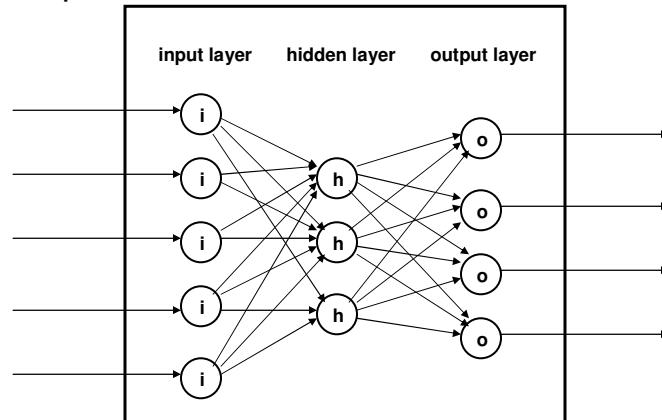
Classification with Neural Networks

24

Back Propagation Network

Example with:

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



28-Aug-08

Classification with Neural Networks

25

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 an high value.

- **Momentum:**

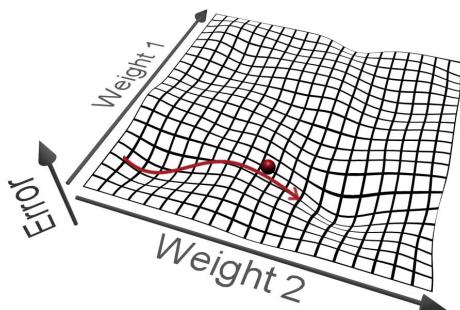
The momentum is zero or a small positive value. This parameter influences the speed of convergence of the learning process and specifies which fraction of the "learn effect" of the previous training sample should be added to the "learn effect" of the next training sample.

28-Aug-08

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

28-Aug-08

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:**
In theory one hidden layer can learn “all problems”, but the use of a second hidden layer may speed up the learning process. Advice: start first with an empty 2nd hidden layer.

28-Aug-08

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]

28-Aug-08

Classification with Neural
Networks

29

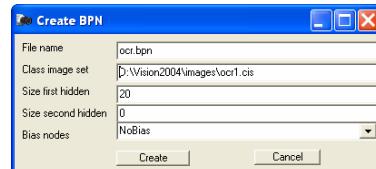
Demonstration Training BPN

- Create new BPN ocr.bpn (File menu),
use ocr1.cis and size first hidden layer = 20, second = 0, NoBias
- Go to Training form
- 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

28-Aug-08

Classification with Neural
Networks

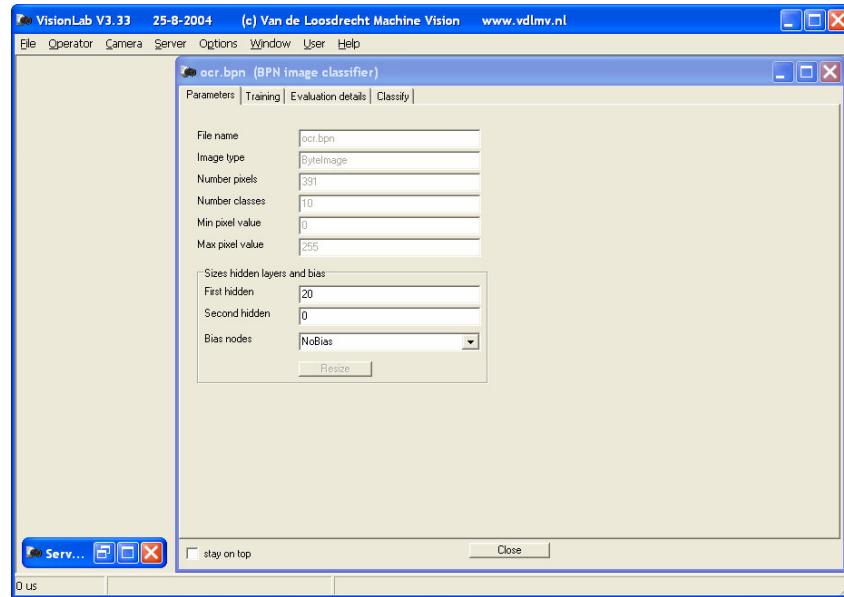
30

Demonstration Training BPN

28-Aug-08

Classification with Neural Networks

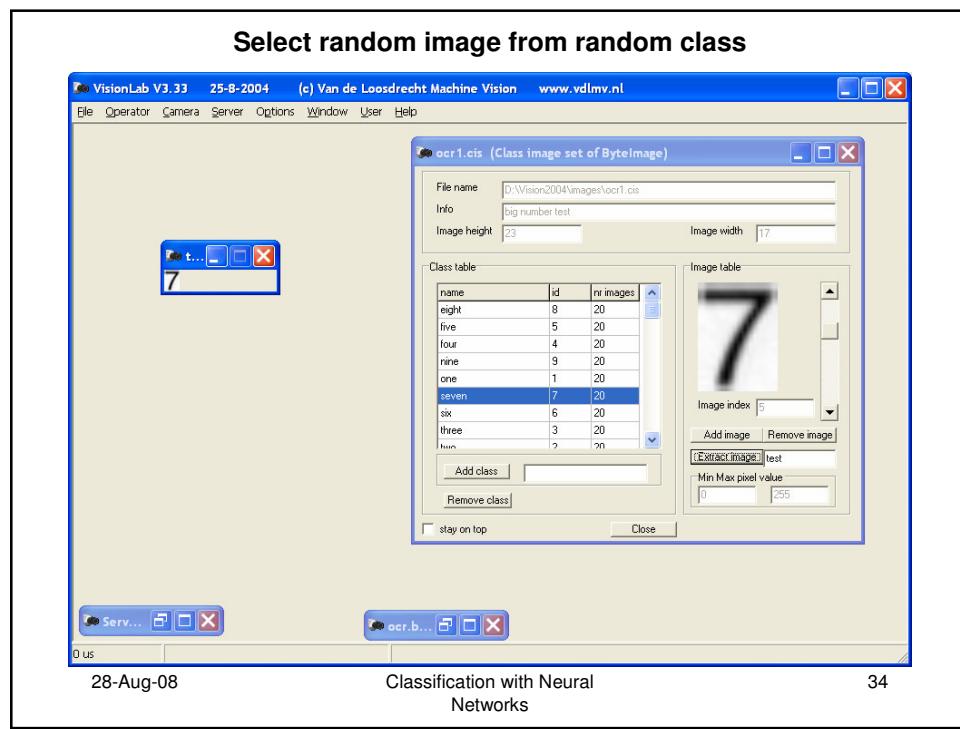
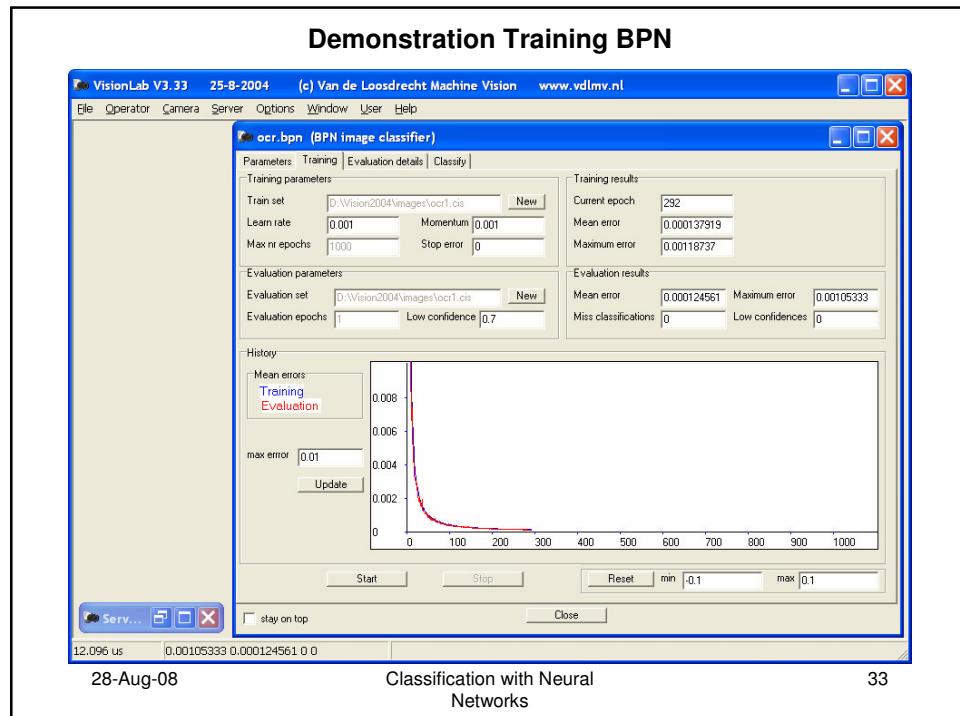
31

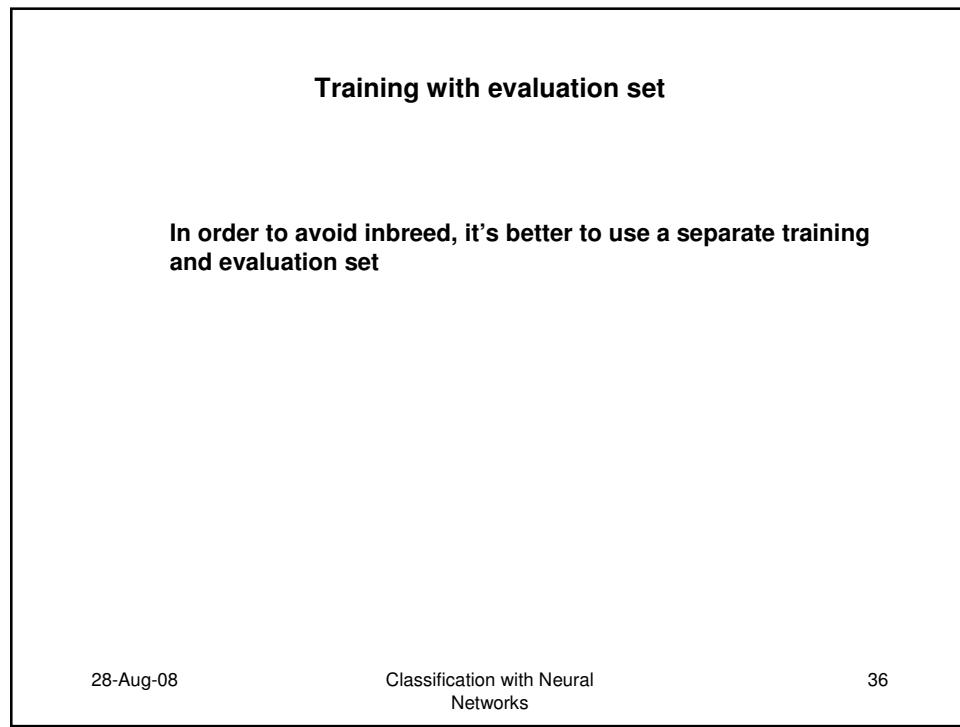
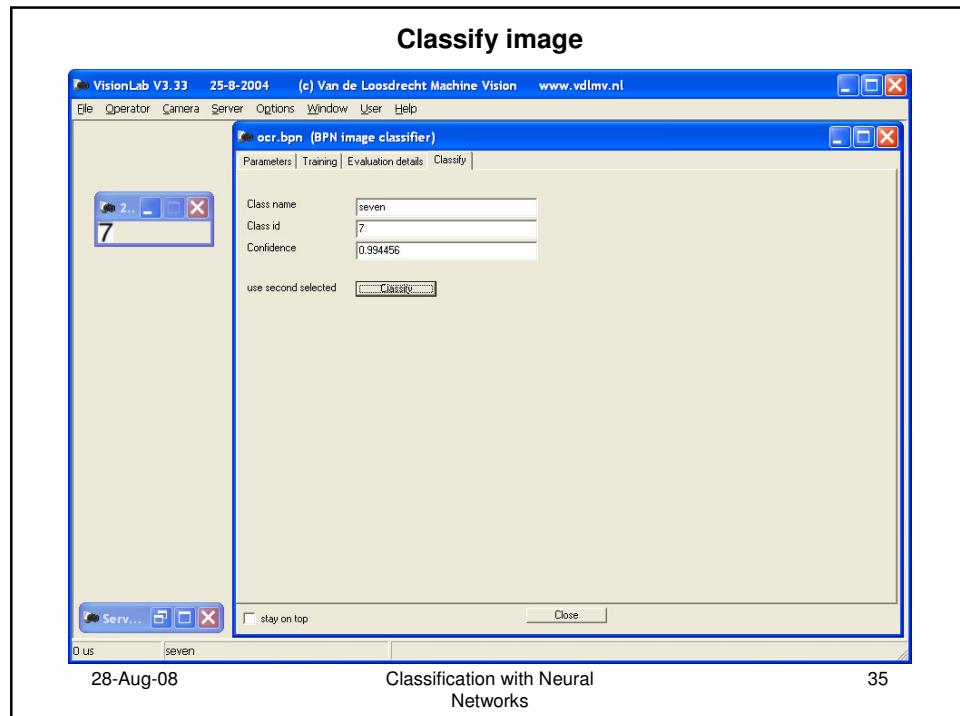
Demonstration Training BPN

28-Aug-08

Classification with Neural Networks

32





Demonstration Training BPN with evaluation set

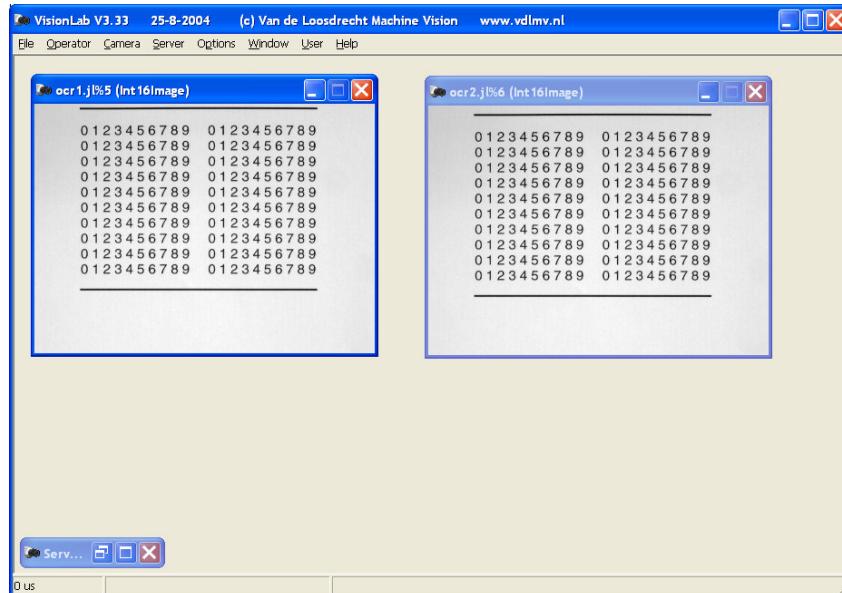
- Create new CIS and some classes (no slides)
- Open image ocr1.jl and ocr2.jl, same object slightly different position and lighting condition
- Use script gen_ocr2_cis.jls to generate ocr2.cis
- Create new BPN ocr2.bpn (File menu), use ocr1.cis and size first hidden layer = 20, second = 0, NoBias
- Go to Training form
- Start training, and select as evaluation set ocr2.cis
- 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

28-Aug-08

Classification with Neural Networks

37

Compare image ocr1.jl and ocr2.jl

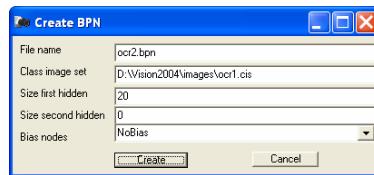


28-Aug-08

Classification with Neural Networks

38

Demonstration Training BPN with evaluation set

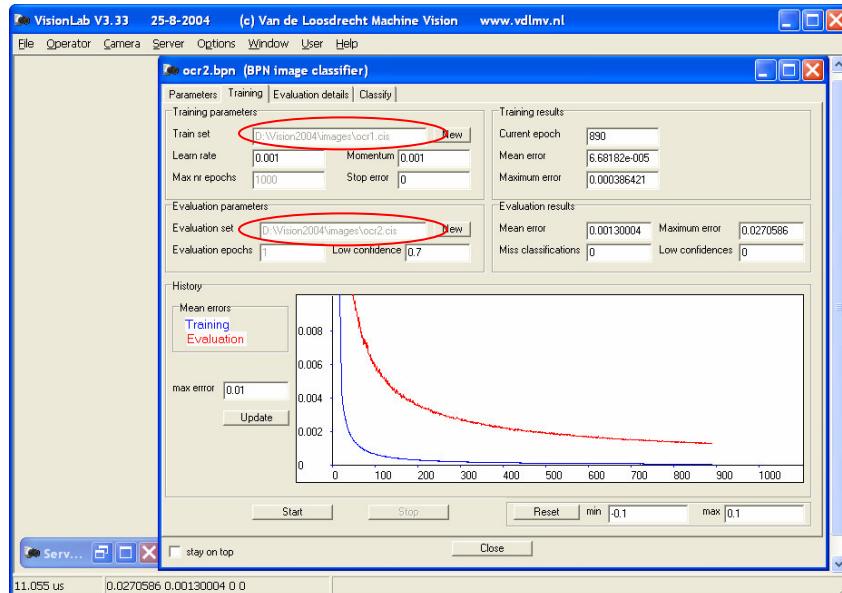


28-Aug-08

Classification with Neural Networks

39

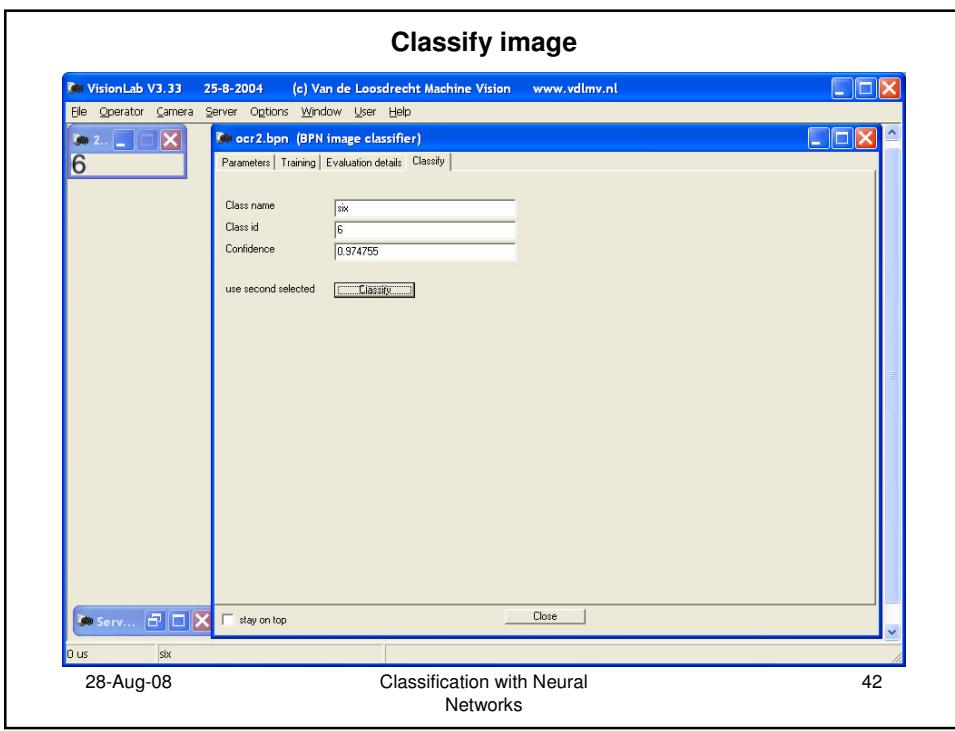
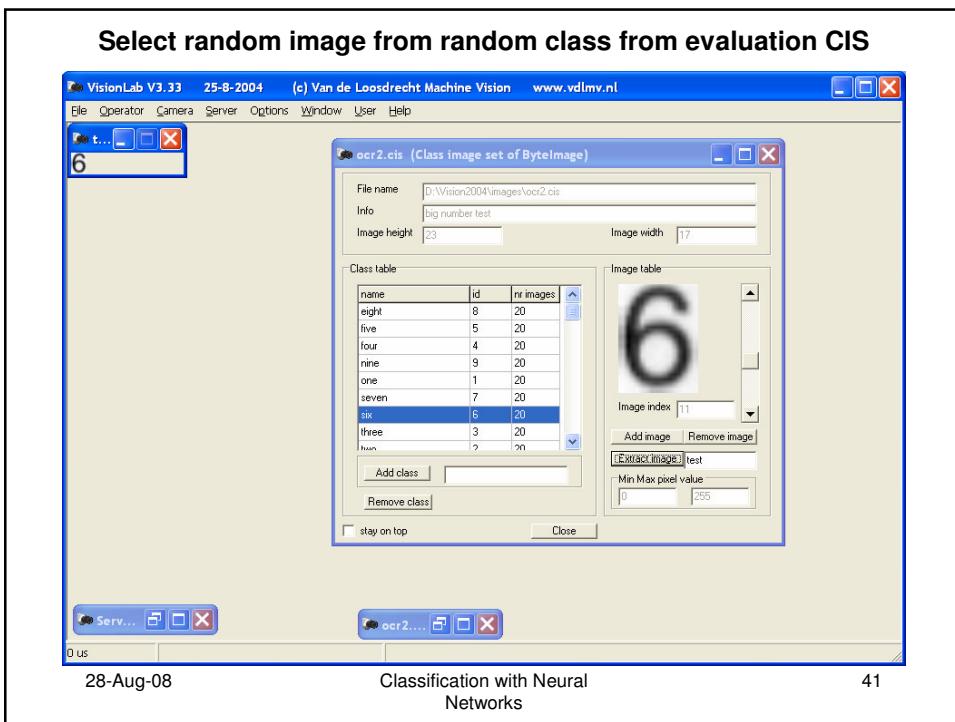
Demonstration Training BPN with evaluation set



28-Aug-08

Classification with Neural Networks

40



Demo analyzing learning problems (*)

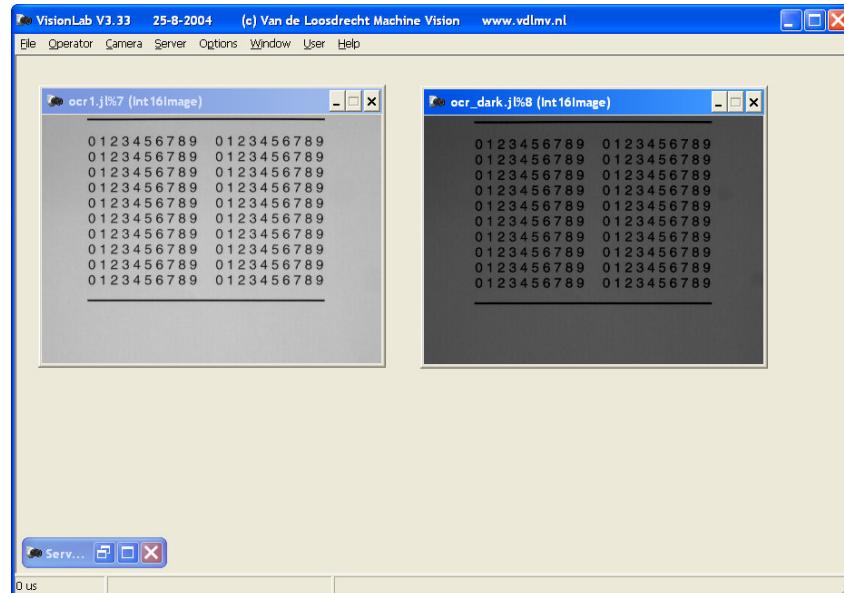
- Compare images `ocr1.jl` and `ocr_dark.jl` with LUT clip
- Change evaluation set to `ocr_dark.cis`,
this is generated from image `ocr_dark.jl` under very low lighting conditions
- Reset network with -0.5 and 0.5. ($[-0.1 \dots 0.1]$ will give better results)
- Start training and stop after 12 epochs,
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'

28-Aug-08

Classification with Neural Networks

43

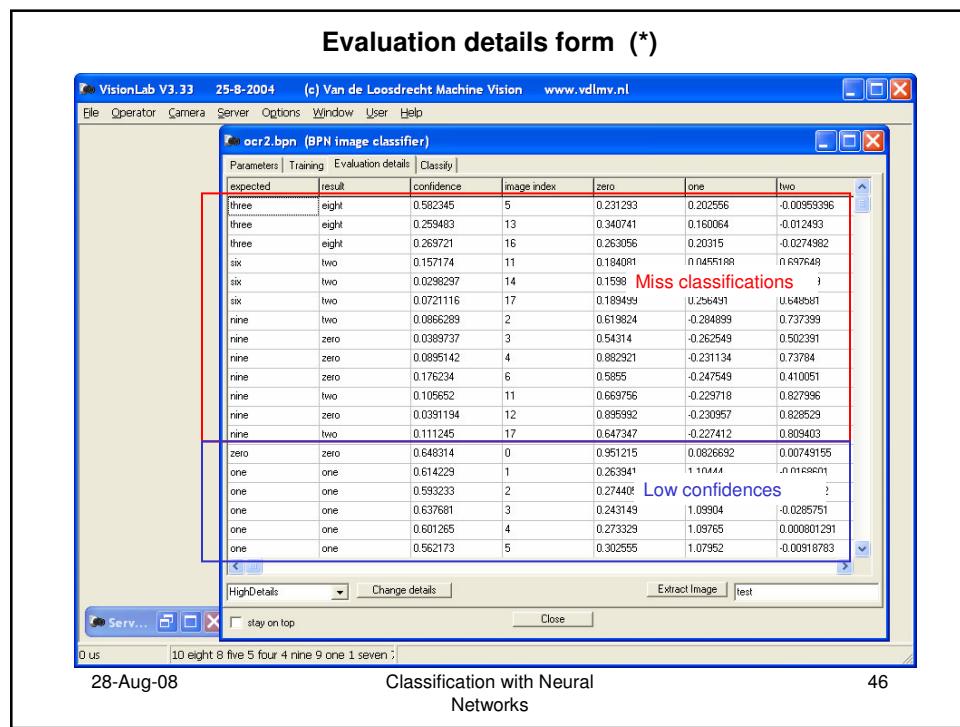
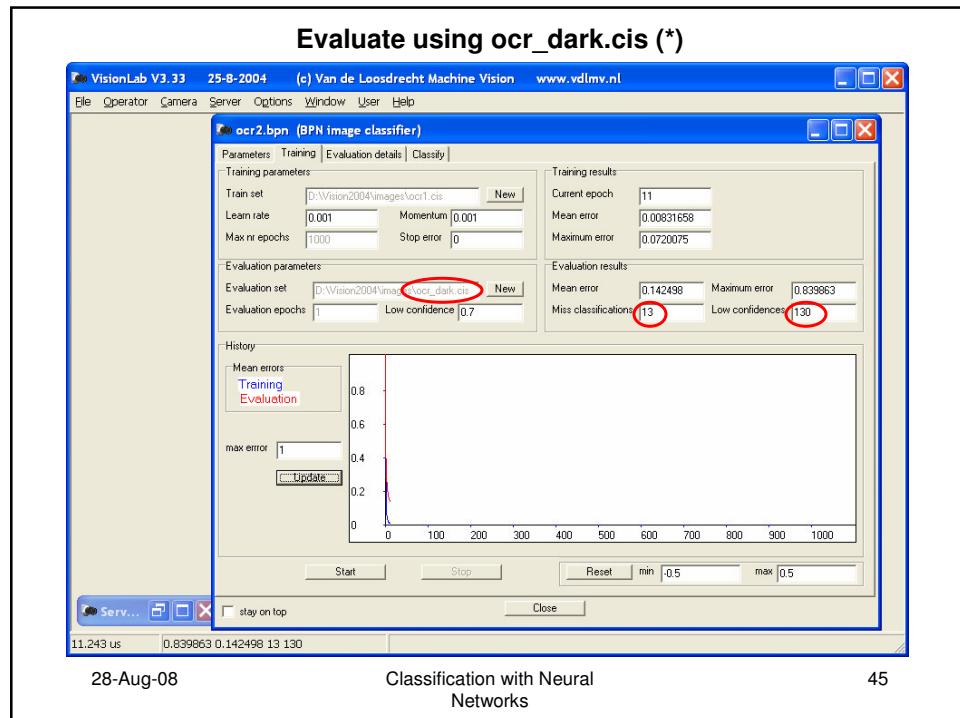
Comparing `ocr1.jl` and `ocr_dark.jl` (*)



28-Aug-08

Classification with Neural Networks

44



Extract offending image (*)

VisionLab V3.33 25-8-2004 (c) Van de Loosdrecht Machine Vision www.vdlmv.nl

File Operator Camera Server Options Window User Help

ocr2.bpn (BPN image classifier)

Parameters Training Evaluation details Classify

expected	result	confidence	image index	zero	one	two
three	eight	0.592345	5	0.231293	0.202556	-0.00953936
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	0.697648
six	two	0.0298297	14	0.159878	0.120233	0.609769
six	two	0.0721116	17	0.189499	0.256491	0.648581
nine	two	0.0865289	2	0.619824	-0.284899	0.737399
nine	zero	0.0389737	3	0.54314	-0.262549	0.502391
nine	zero	0.0895142	4	0.0882921	-0.231134	0.737384
nine	zero	0.176234	6	0.5855	-0.247549	0.410051
nine	two	0.105652	11	0.668756	-0.229718	0.827396
nine	zero	0.0391194	12	0.895992	-0.230957	0.828529
nine	two	0.111245	17	0.647347	-0.227412	0.809403
zero	zero	0.648314	0	0.951215	0.0826562	0.00749155
one	one	0.614223	1	0.263941	1.10444	-0.0168601
one	one	0.93233	2	0.274405	1.0748	-0.0281802
one	one	0.637681	3	0.243149	1.05904	-0.0285751
one	one	0.601265	4	0.273329	1.05765	0.000801291
one	one	0.562173	5	0.302555	1.07952	-0.00918783

Extract image miss

28-Aug-08 Classification with Neural Networks 47

Examine high details (*)

VisionLab V3.33 25-8-2004 (c) Van de Loosdrecht Machine Vision www.vdlmv.nl

File Operator Camera Server Options Window User Help

ocr2.bpn (BPN image classifier)

Parameters Training Evaluation details Classify

expected	result	confidence	image index	zero	one	two
three	eight	0.592345	5	0.231293	0.202556	-0.00953936
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	0.697648
six	two	0.0298297	14	0.159878	0.120233	0.609769
six	two	0.0721116	17	0.189499	0.256491	0.648581
nine	two	0.0865289	2	0.619824	-0.284899	0.737399
nine	zero	0.0389737	3	0.54314	-0.262549	0.502391
nine	zero	0.0895142	4	0.0882921	-0.231134	0.737384
nine	zero	0.176234	6	0.5855	-0.247549	0.410051
nine	two	0.105652	11	0.668756	-0.229718	0.827396
nine	zero	0.0391194	12	0.895992	-0.230957	0.828529
nine	two	0.111245	17	0.647347	-0.227412	0.809403
zero	zero	0.648314	0	0.951215	0.0826562	0.00749155
one	one	0.614223	1	0.263941	1.10444	-0.0168601
one	one	0.93233	2	0.274405	1.0748	-0.0281802
one	one	0.637681	3	0.243149	1.05904	-0.0285751
one	one	0.601265	4	0.273329	1.05765	0.000801291
one	one	0.562173	5	0.302555	1.07952	-0.00918783

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

Low confidences
0: 0.263941
1: 1.10444

Extract image miss

28-Aug-08 Classification with Neural Networks 48

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

28-Aug-08 Classification with Neural Networks 49

Exercise OCR dark (*)

- Experiment with learning parameters to get better results

28-Aug-08 Classification with Neural Networks 50

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

28-Aug-08

Classification with Neural Networks

51

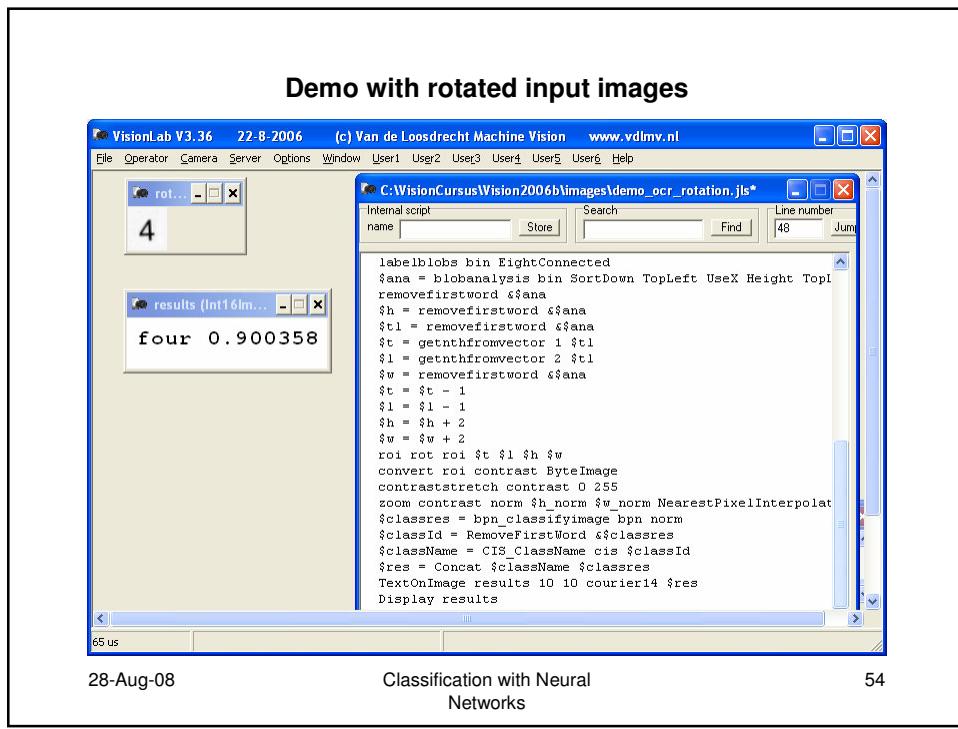
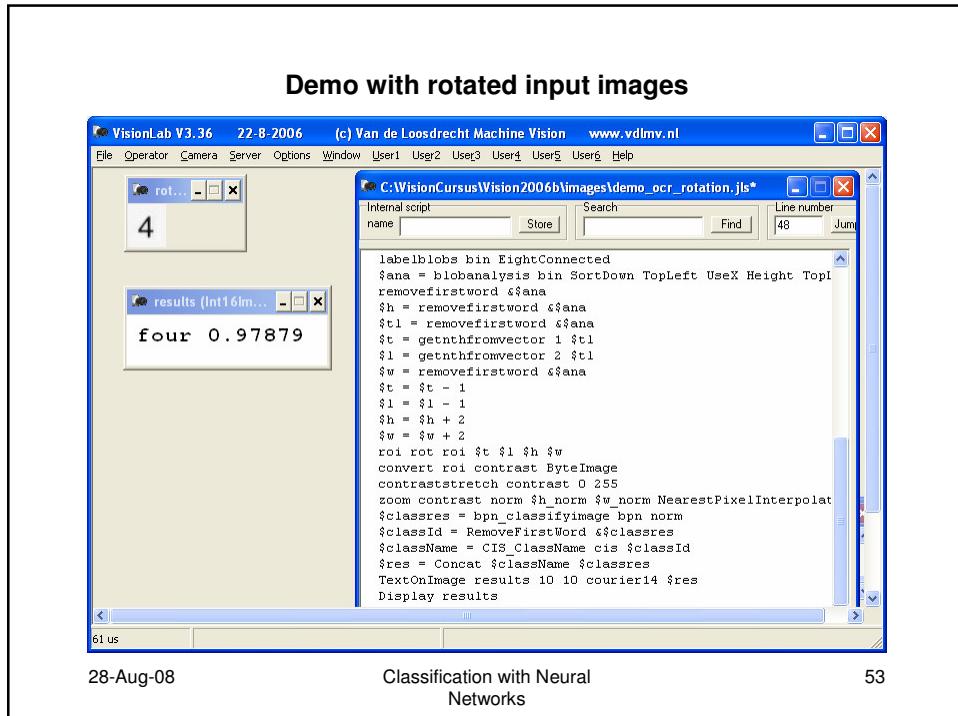
Demo robustness of BPN

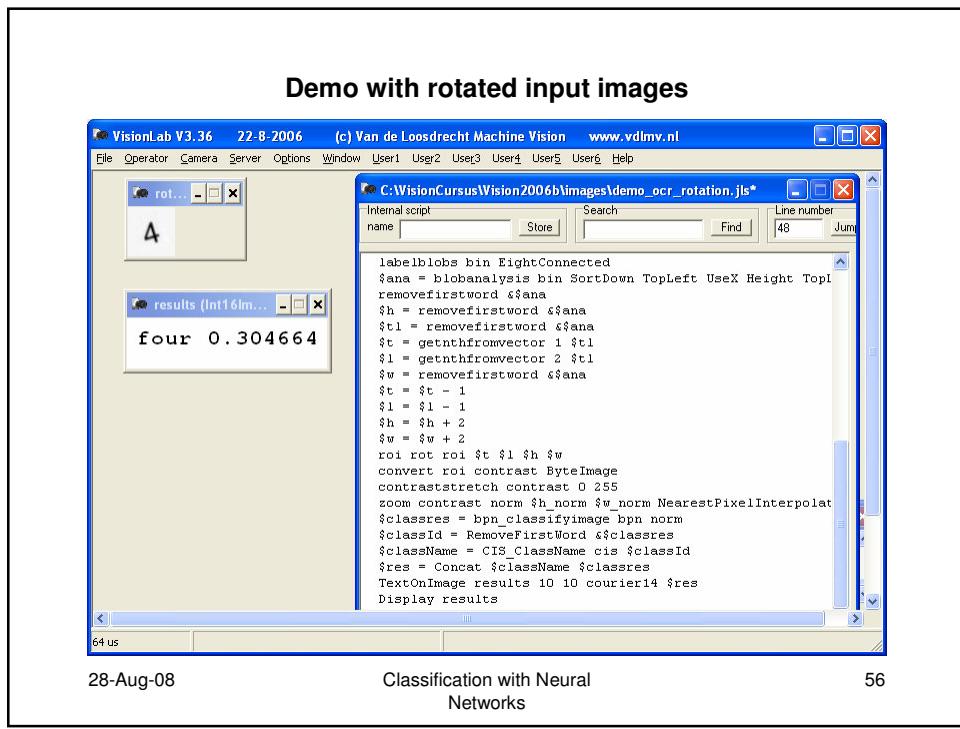
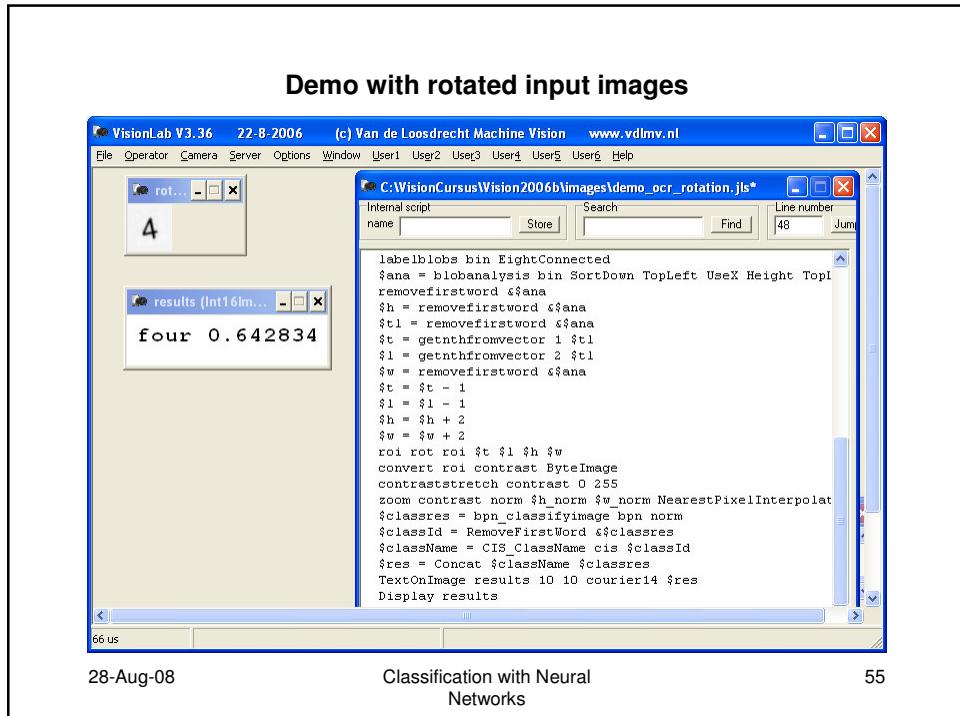
- Execute script `demo_ocr_rotation.jls`
- Execute script `demo_ocr_size.jls`
- Execute script `demo_ocr_contrast.jls`

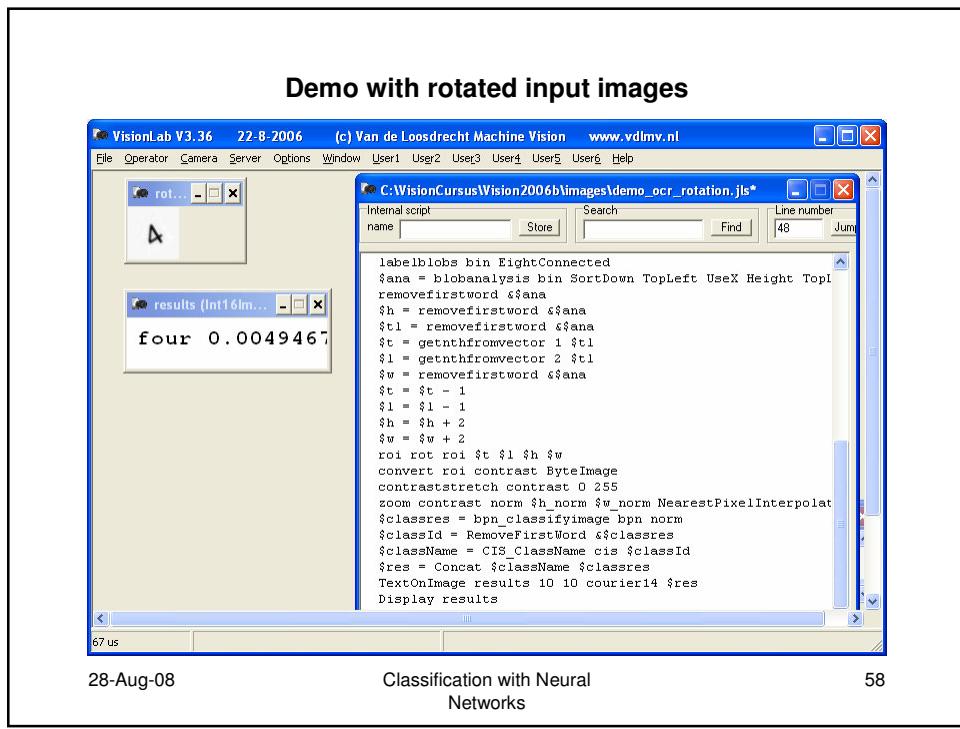
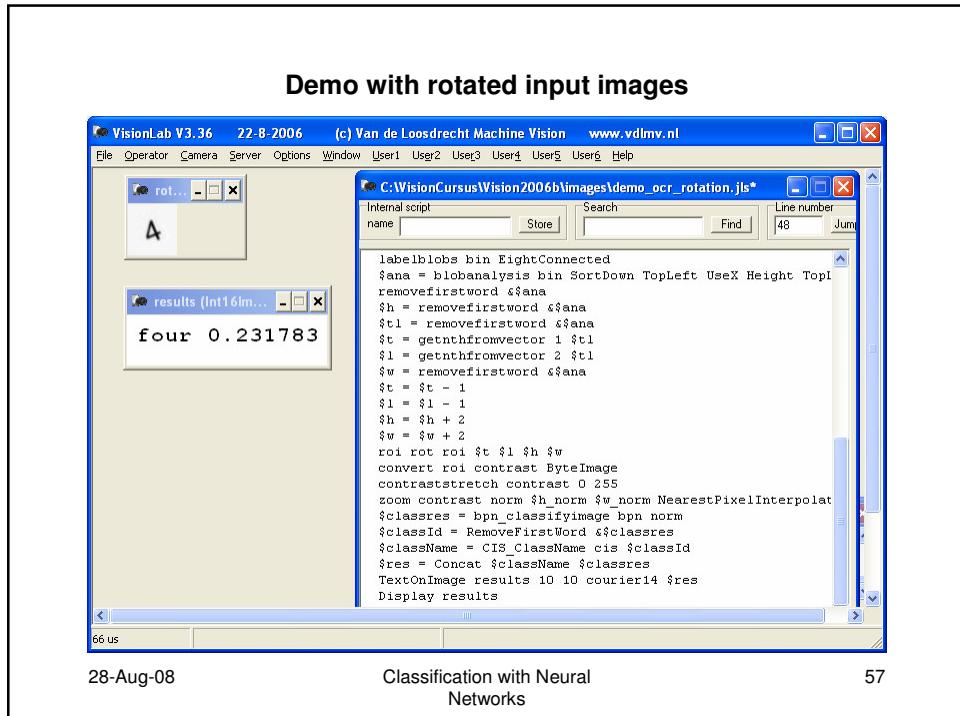
28-Aug-08

Classification with Neural Networks

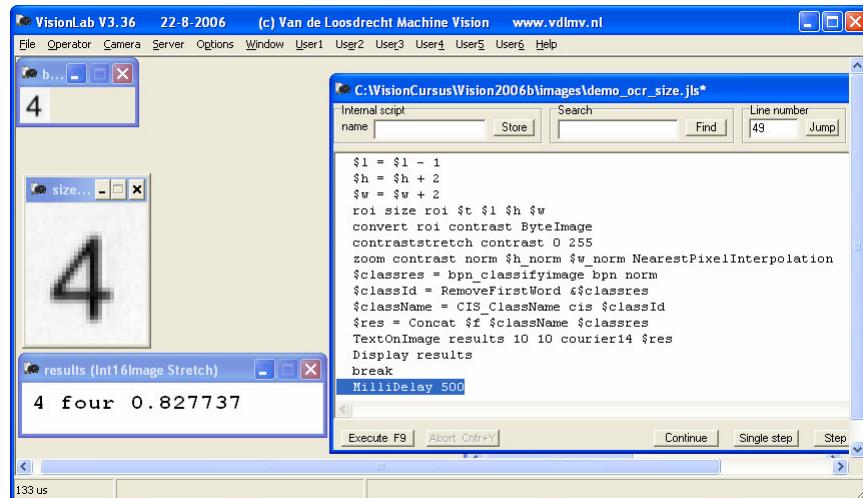
52







Demo with zoomed input images

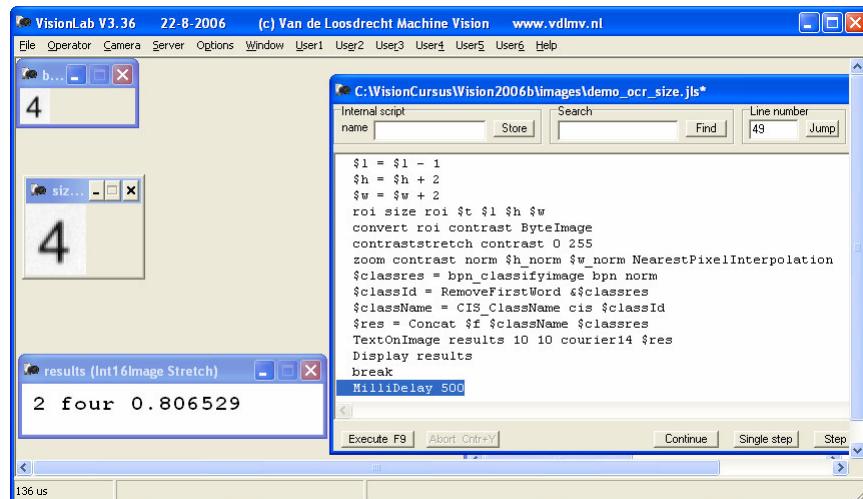


28-Aug-08

Classification with Neural Networks

59

Demo with zoomed input images

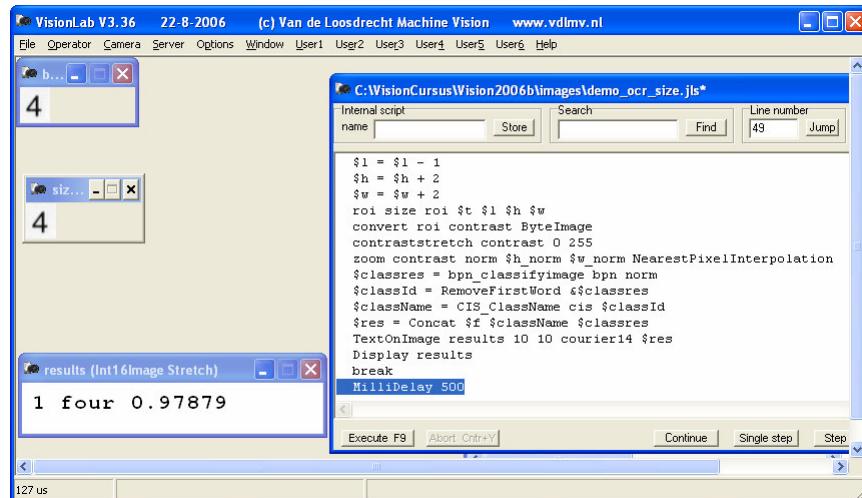


28-Aug-08

Classification with Neural Networks

60

Demo with zoomed input images



The screenshot shows the VisionLab V3.36 interface. The main window displays a zoomed-in image of the digit '4'. Below the image, the results window shows the output: '1 four 0.97879'. The script window contains the following code:

```

$1 = $1 - 1
$h = $h + 2
$w = $w + 2
roi size roi $t $1 $h $w
convert roi contrast ByteImage
contraststretch contrast 0 255
zoom contrast norm $h_norm $w_norm NearestPixelInterpolation
$classes = bpn_classifyimage bpn norm
$classId = RemoveFirstWord $classes
$className = CIS_ClassName cis $classId
$res = Concat $className $classes
TextOnImage results 10 10 courier14 $res
Display results
break
MilliDelay 500

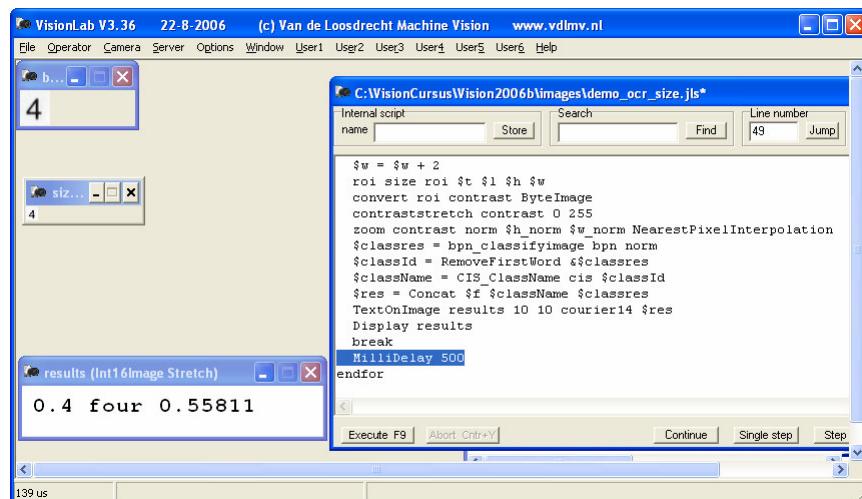
```

28-Aug-08

Classification with Neural Networks

61

Demo with zoomed input images



The screenshot shows the VisionLab V3.36 interface. The main window displays a zoomed-in image of the digit '4'. Below the image, the results window shows the output: '0.4 four 0.55811'. The script window contains the following code:

```

$w = $w + 2
roi size roi $t $1 $h $w
convert roi contrast ByteImage
contraststretch contrast 0 255
zoom contrast norm $h_norm $w_norm NearestPixelInterpolation
$classes = bpn_classifyimage bpn norm
$classId = RemoveFirstWord $classes
$className = CIS_ClassName cis $classId
$res = Concat $className $classes
TextOnImage results 10 10 courier14 $res
Display results
break
MilliDelay 500
endfor

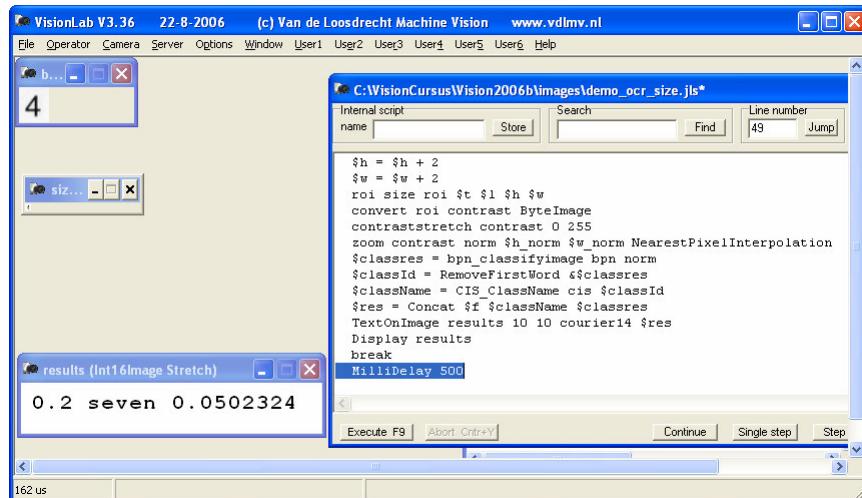
```

28-Aug-08

Classification with Neural Networks

62

Demo with zoomed input images

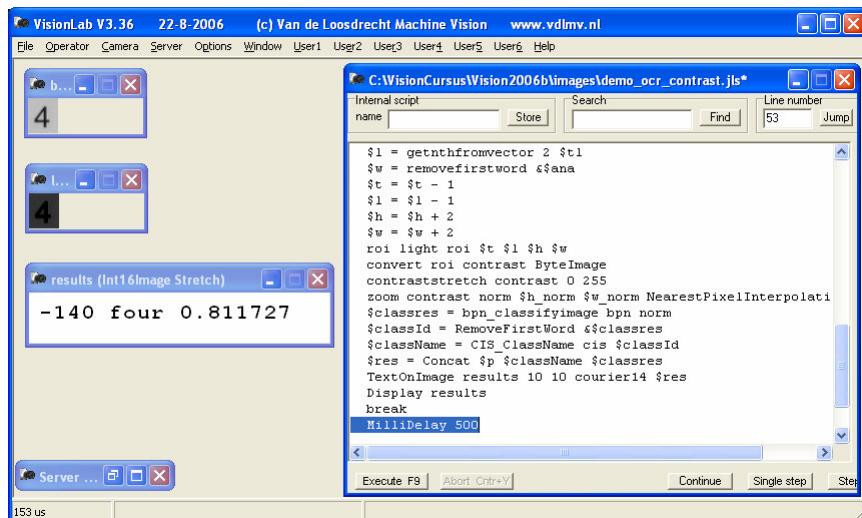


28-Aug-08

Classification with Neural Networks

63

Demo with changed light conditions

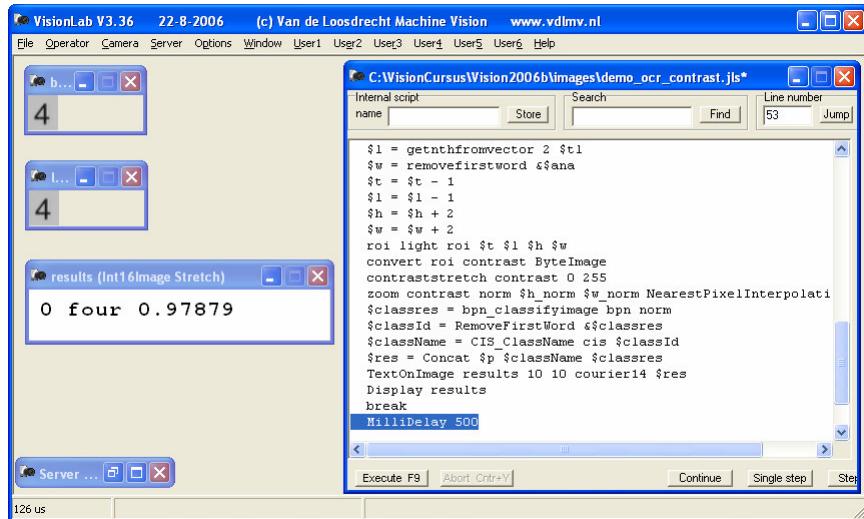


28-Aug-08

Classification with Neural Networks

64

Demo with changed light conditions

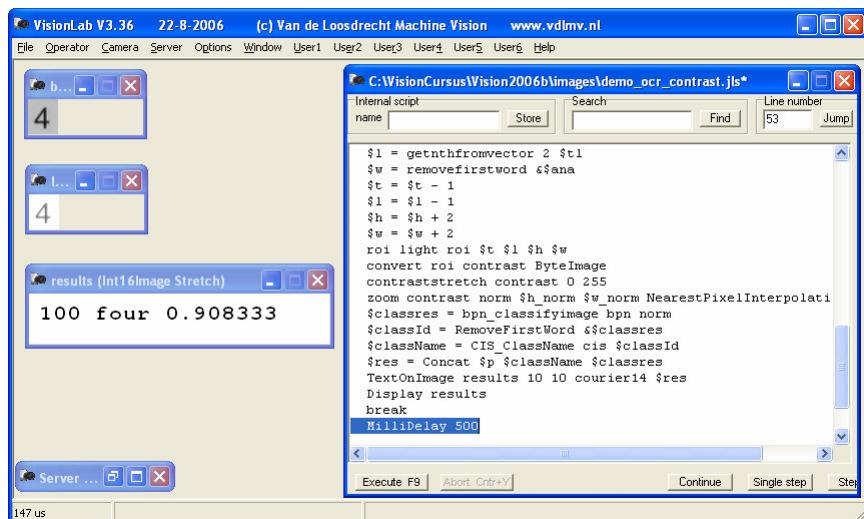


28-Aug-08

Classification with Neural Networks

65

Demo with changed light conditions

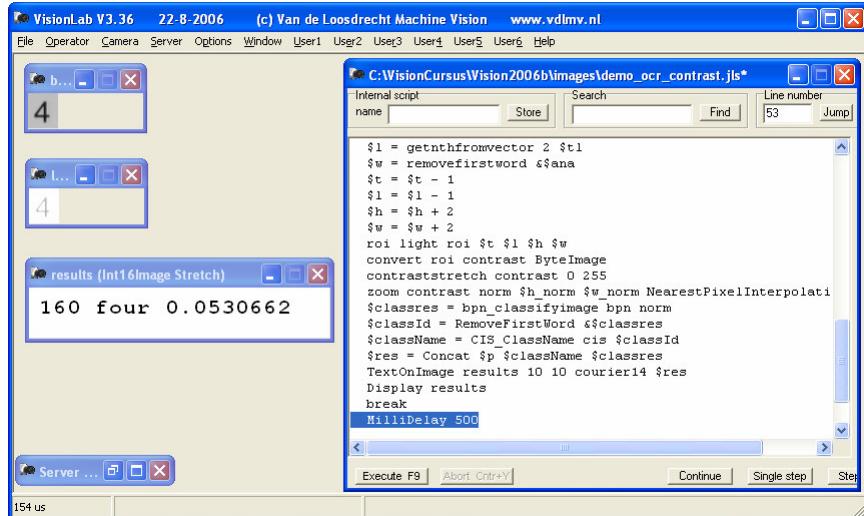


28-Aug-08

Classification with Neural Networks

66

Demo with changed light conditions



28-Aug-08

Classification with Neural Networks

67

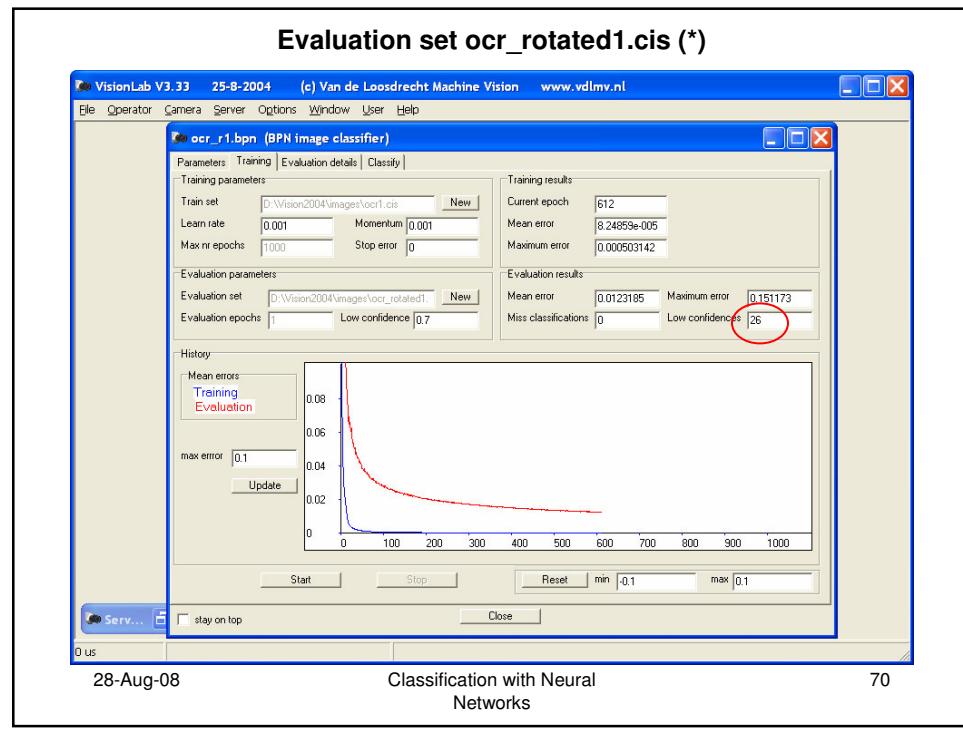
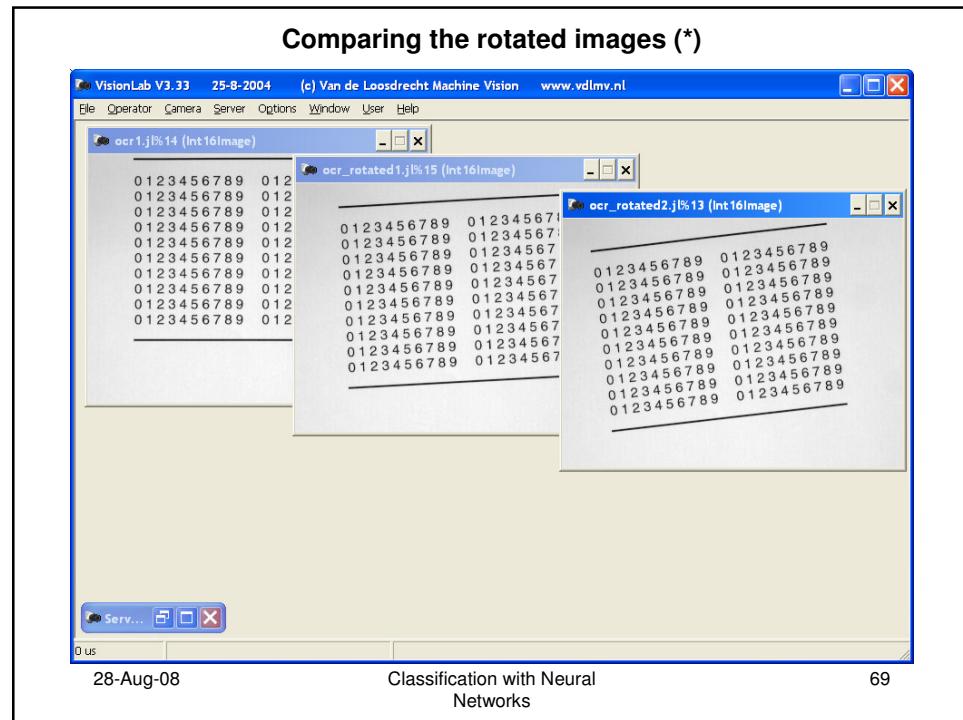
Demo with wrong evaluation set (*)

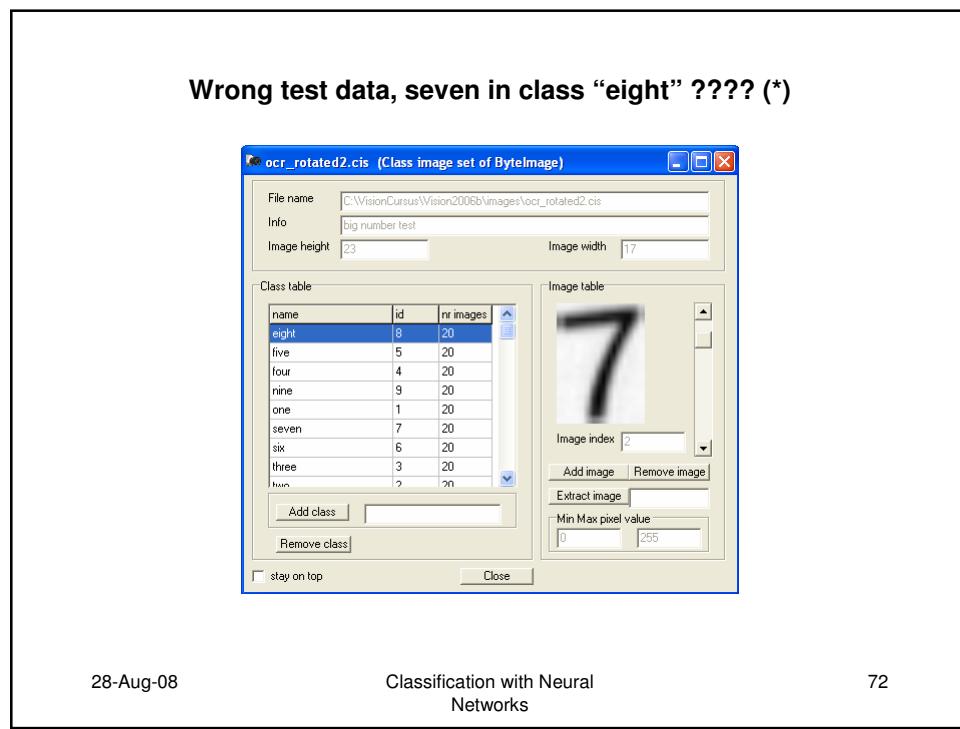
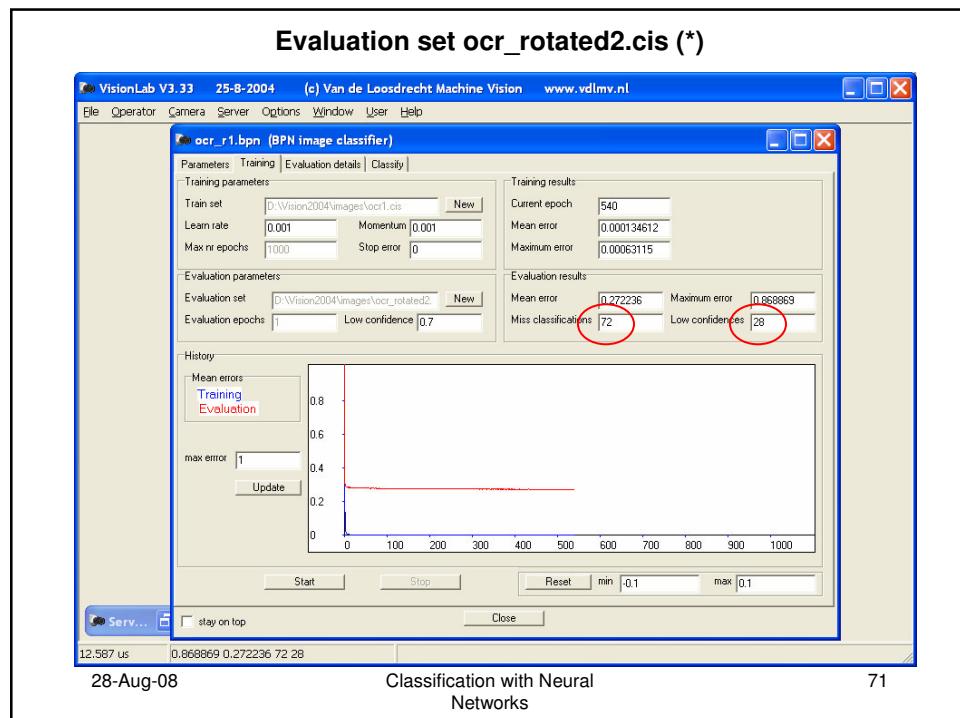
- The BPN was trained with numbers without rotation and is expected to perform worse if the digits are rotated
- Compare test images `ocr1.jl`, `ocr_rotated1.jl` and `ocr_rotated2.jl`
- For the rotated image `ocr1` a CIS is generated with `gen_ocr_rotated1_cis.jls`
- For the rotated image `ocr2` a CIS 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 !

28-Aug-08

Classification with Neural Networks

68





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

28-Aug-08

Classification with Neural Networks

73

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`,

28-Aug-08

Classification with Neural Networks

74

Robustness OCR (*)

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

This is because the images are normalized for these aspects

28-Aug-08

Classification with Neural
Networks

75

Feature vector classification (*)

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

28-Aug-08

Classification with Neural
Networks

76

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

28-Aug-08

Classification with Neural Networks

77

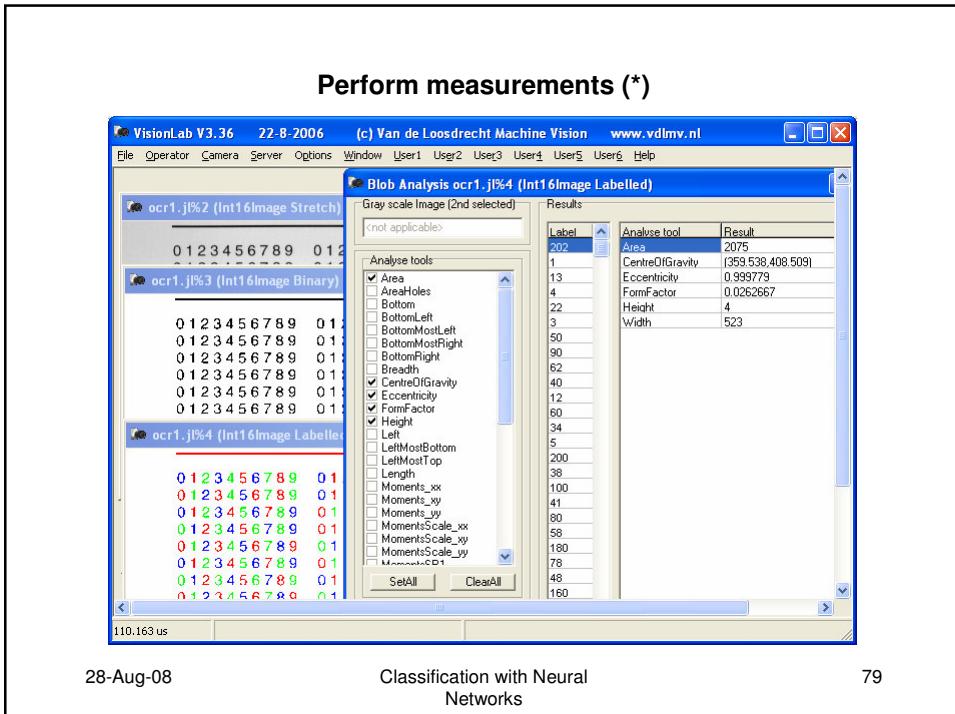
Demo feature classifying (*)

- { Demonstrate how to perform measurements
 - Open ocr1.jl (grayscale image)
 - ThresholdIsodata (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 bias
- Demo learning, no feature: cg.x, cg.y and SumXXX
(use keyboard space to de-select 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 cg.x to distinguish between 6 and 9
 - 6: $7.3 < \text{cg.x} < 8.1, 10.7 < \text{cg.y} < 11.8$
 - 9: $8.6 < \text{cg.x} < 9.9, 10 < \text{cg.y} < 11.2$

28-Aug-08

Classification with Neural Networks

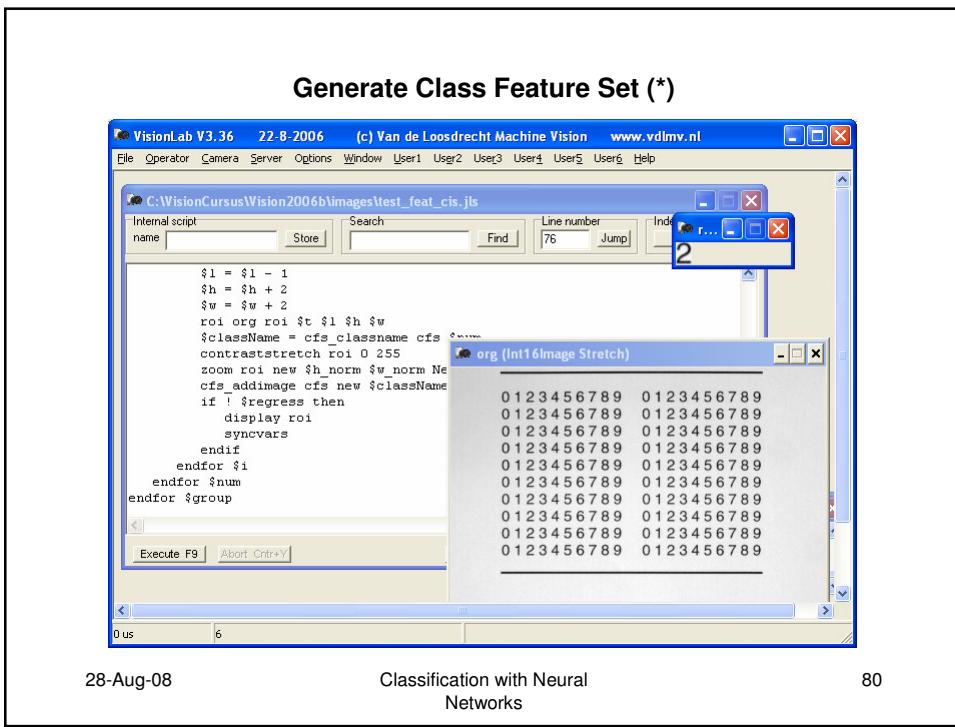
78



28-Aug-08

Classification with Neural Networks

79



28-Aug-08

Classification with Neural Networks

80

Class Feature Set (CFS) (*)

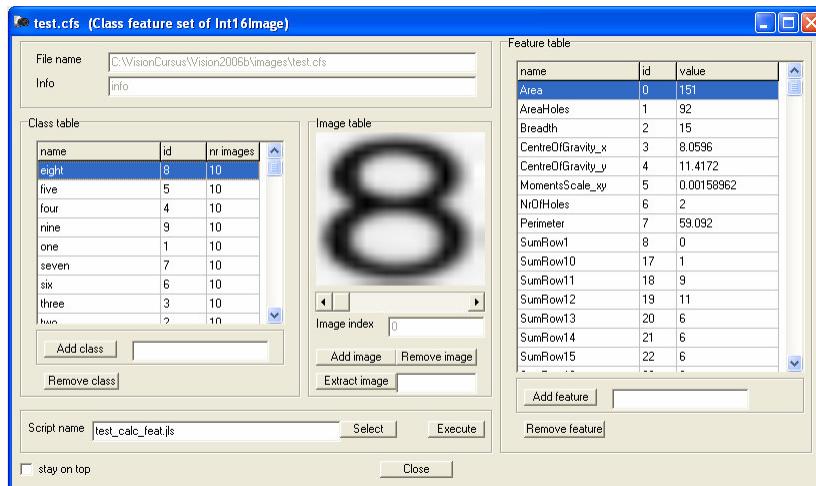
- A CFS is a collection of images with their associated classes.
All images in a CFS must have the same image type
- CFS:
 - Class table with the name and id of each class
 - For each class an image table,
each image in an image table has its unique image index number
 - For each image a feature table,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

28-Aug-08

Classification with Neural Networks

81

Class Feature Set (*)

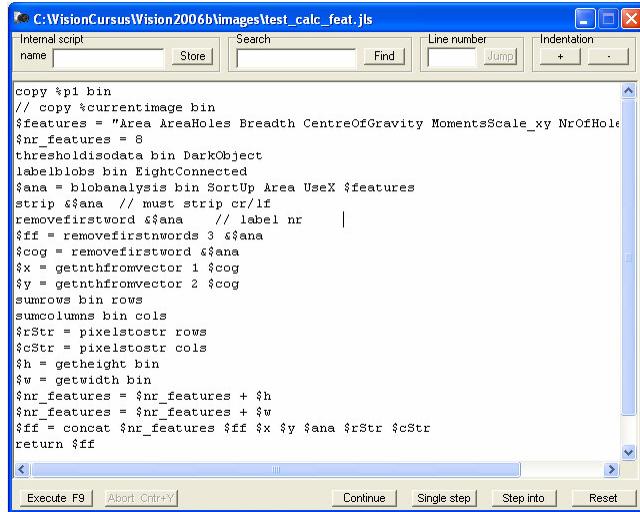


28-Aug-08

Classification with Neural Networks

82

Script for calculating features (*)



```

copy %p1 bin
// copy %currentimage bin
$features = "Area AreaHoles Breadth CentreOfGravity MomentsScale_xy NrOfHoles
$nr_features = 8
thresholdisodata bin DarkObject
labelblobs bin EightConnected
$ana = blobanalysis bin SortUp Area UseX $features
strip $ana // must strip cr/lf
removefirstword $ana // label nr
$ff = removefirstword $ana
$cog = removefirstword $ana
$x = getnthfromvector 1 $cog
$y = getnthfromvector 2 $cog
sumrows bin rows
sumcolumns bin cols
$rStr = pixelstostr rows
$cStr = pixelstostr cols
$h = getheight bin
$w = getwidth bin
$nr_features = $nr_features + $h
$nr_features = $nr_features + $w
$ff = concat $nr_features $ff $x $y $ana $rStr $cStr
return $ff

```

28-Aug-08

Classification with Neural Networks

83

Conventions for script used for CFS (*)

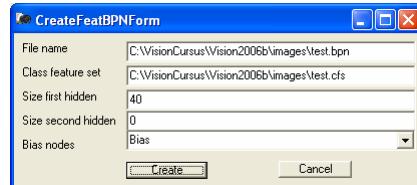
- **Script is used for calculating the features for each image in the CFS**
- **First parameter (%p1) is name of image**
- **Function result is string with features separated by a space and in order of feature id**

28-Aug-08

Classification with Neural Networks

84

Create BPN Feature Classifier (*)

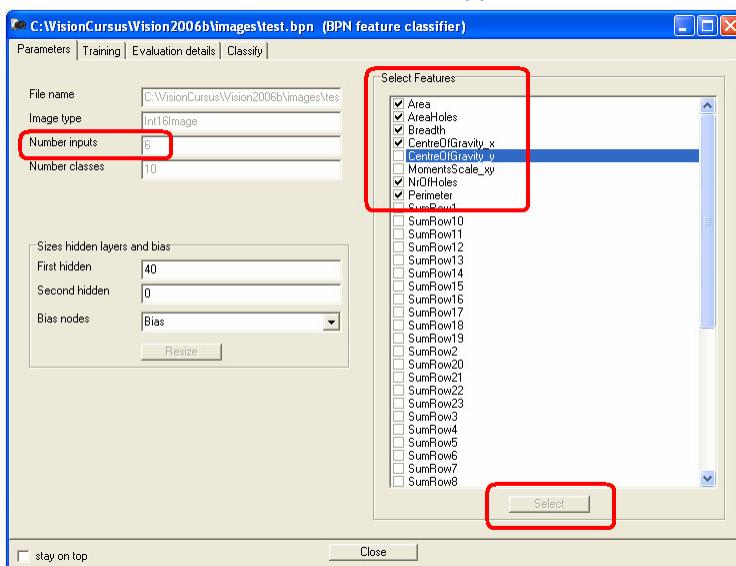


28-Aug-08

Classification with Neural Networks

85

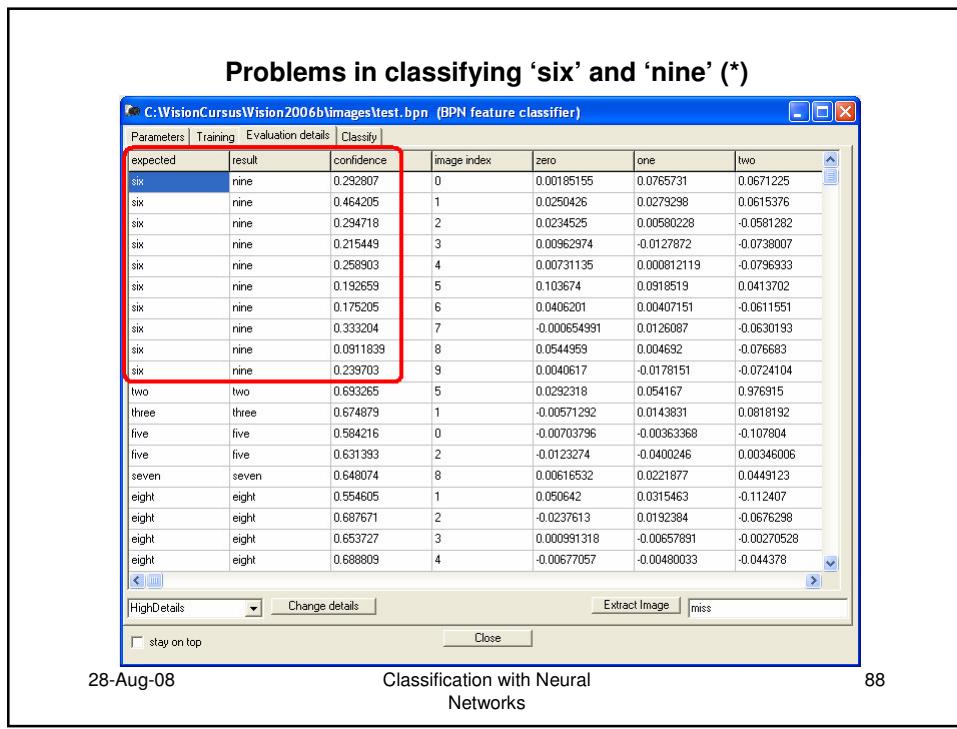
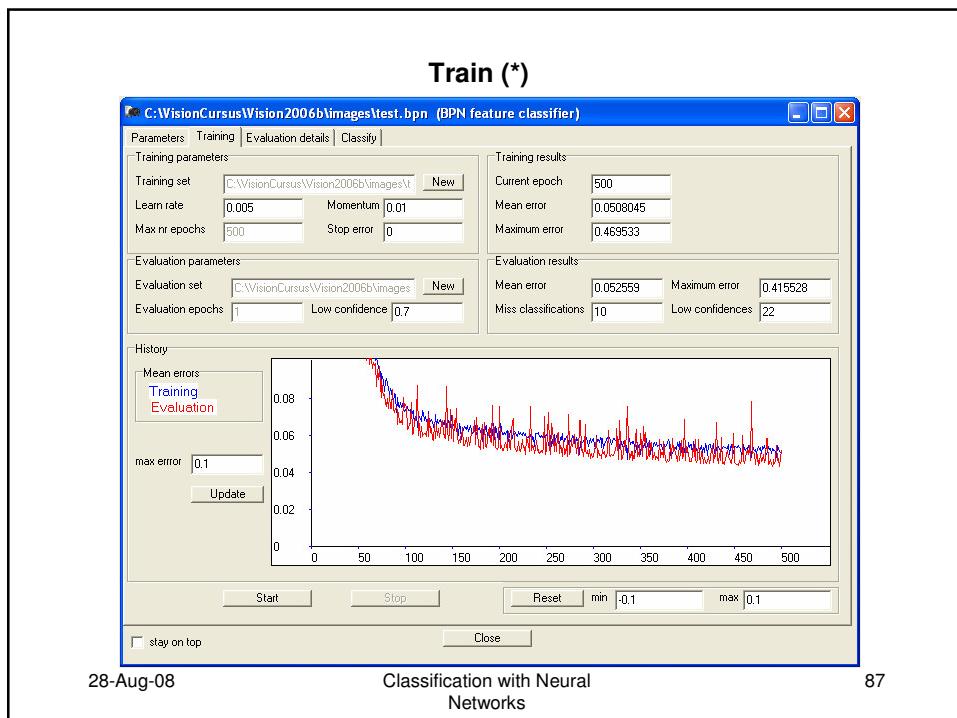
Select Features (*)



28-Aug-08

Classification with Neural Networks

86



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

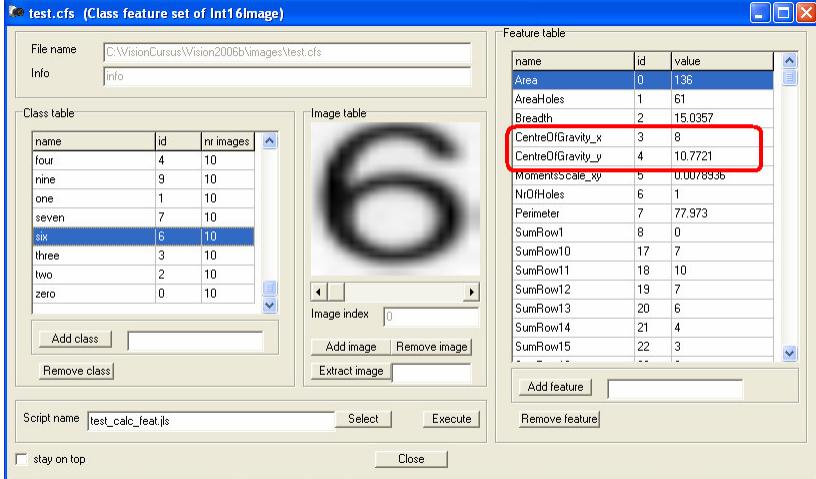
28-Aug-08 Classification with Neural Networks 89

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' : $7.3 < \text{cg.x} < 8.1$, $10.7 < \text{cg.y} < 11.8$
 - 'nine' : $8.6 < \text{cg.x} < 9.9$, $10 < \text{cg.y} < 11.2$
- **cg.x discriminates between 6 and 9**

28-Aug-08 Classification with Neural Networks 90

Features for 'six' (*)



The screenshot shows the Vision software interface with the following details:

- File name:** C:\VisionCursus\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:** Displays a blurred image of the digit 'six'.
- Feature table:**

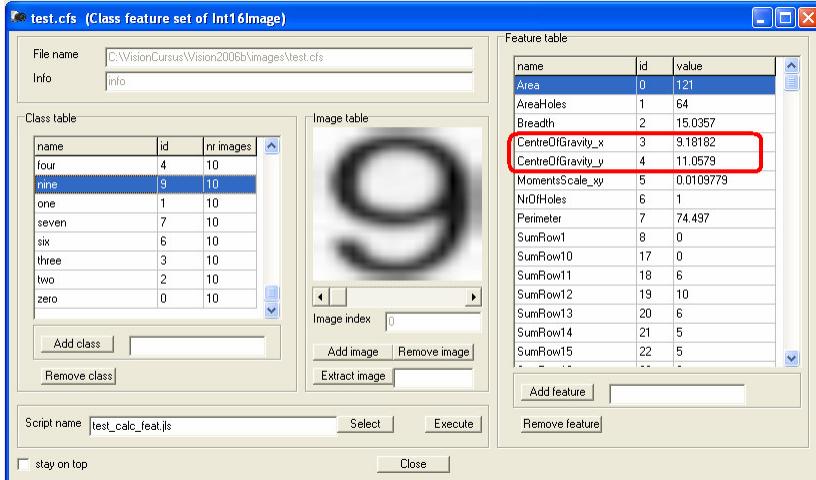
name	id	value
Area	0	136
AreaHoles	1	61
Breadth	2	15.0357
CentreOfGravity_x	3	8
CentreOfGravity_y	4	10.7721
MomentsScale_xy	5	0.0078939
NoHoles	6	1
Perimeter	7	77.973
SumRow1	8	0
SumRow10	17	7
SumRow11	18	10
SumRow12	19	7
SumRow13	20	6
SumRow14	21	4
SumRow15	22	3
- Buttons:** Add class, Remove class, Script name (test_calc_feat.jls), Select, Execute, Close, stay on top.

28-Aug-08

Classification with Neural Networks

91

Features for 'nine' (*)



The screenshot shows the Vision software interface with the following details:

- File name:** C:\VisionCursus\Vision2006b\images\test.cfs
- Info:** info
- Class table:**

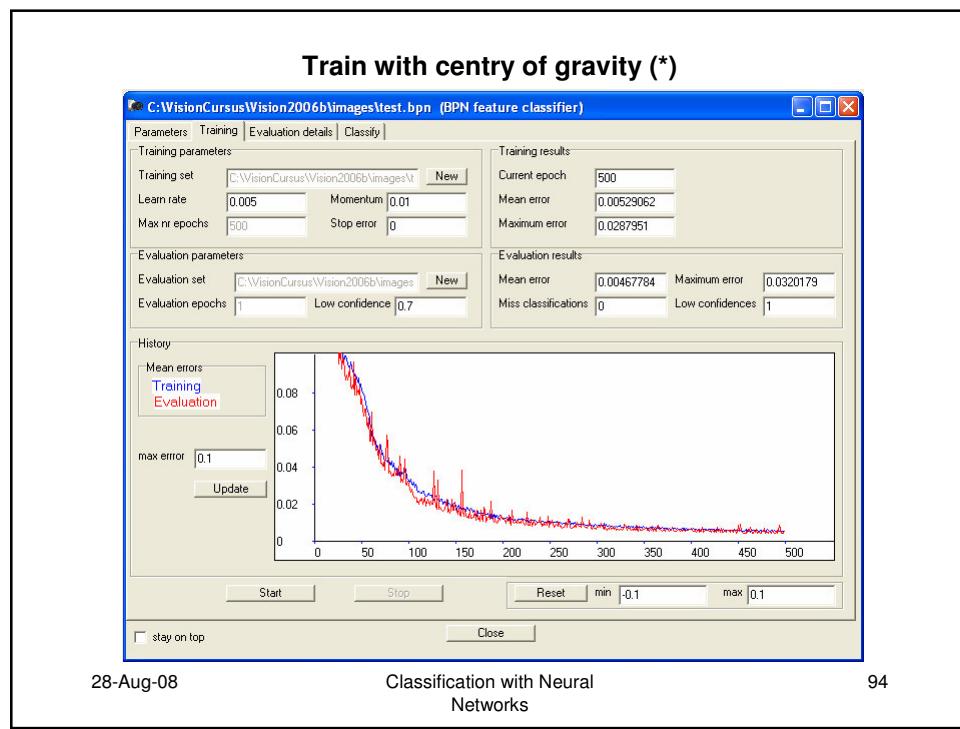
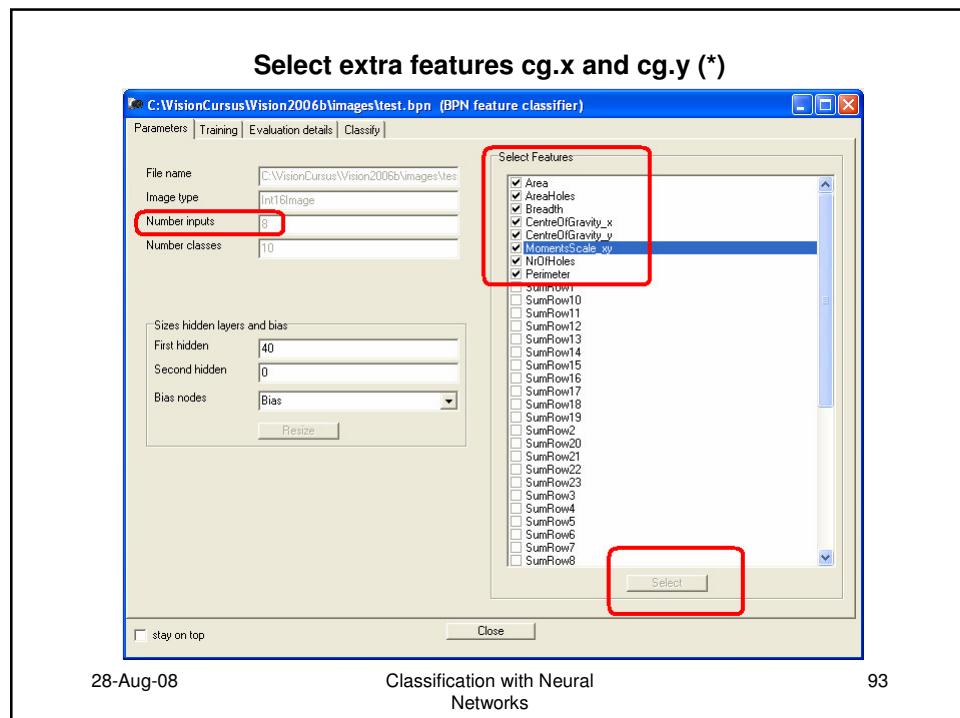
name	id	nr images
four	4	10
nine	9	10
one	1	10
seven	7	10
six	6	10
three	3	10
two	2	10
zero	0	10
- Image table:** Displays a blurred image of the digit 'nine'.
- Feature table:**

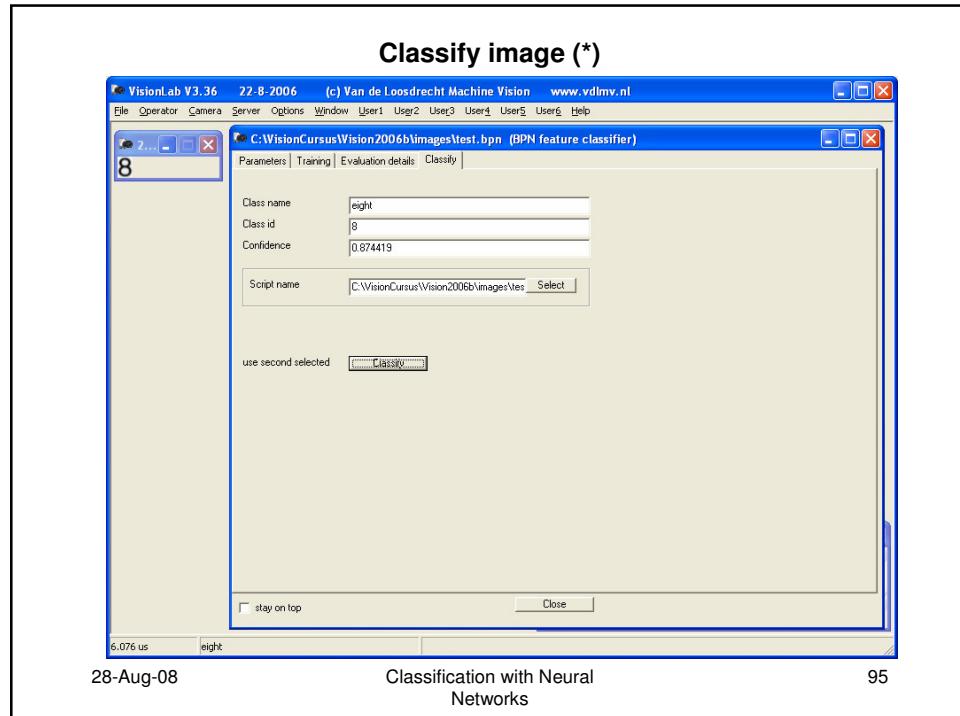
name	id	value
Area	0	121
AreaHoles	1	64
Breadth	2	15.0357
CentreOfGravity_x	3	9.18182
CentreOfGravity_y	4	11.0579
MomentsScale_xy	5	0.0109779
NoHoles	6	1
Perimeter	7	74.497
SumRow1	8	0
SumRow10	17	0
SumRow11	18	6
SumRow12	19	10
SumRow13	20	6
SumRow14	21	5
SumRow15	22	5
- Buttons:** Add class, Remove class, Script name (test_calc_feat.jls), Select, Execute, Close, stay on top.

28-Aug-08

Classification with Neural Networks

92





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

28-Aug-08 Classification with Neural Networks 96

Exercise feature vector classification (*)

Experiment with adding and removing features

Tip: experiment with SumRows and SumCols only