Appendix

Intentionally left blank

Contents

Intentionally left blank

Appendix I

Neocognitron Simulator

1.1 Neocognitron simulator overview

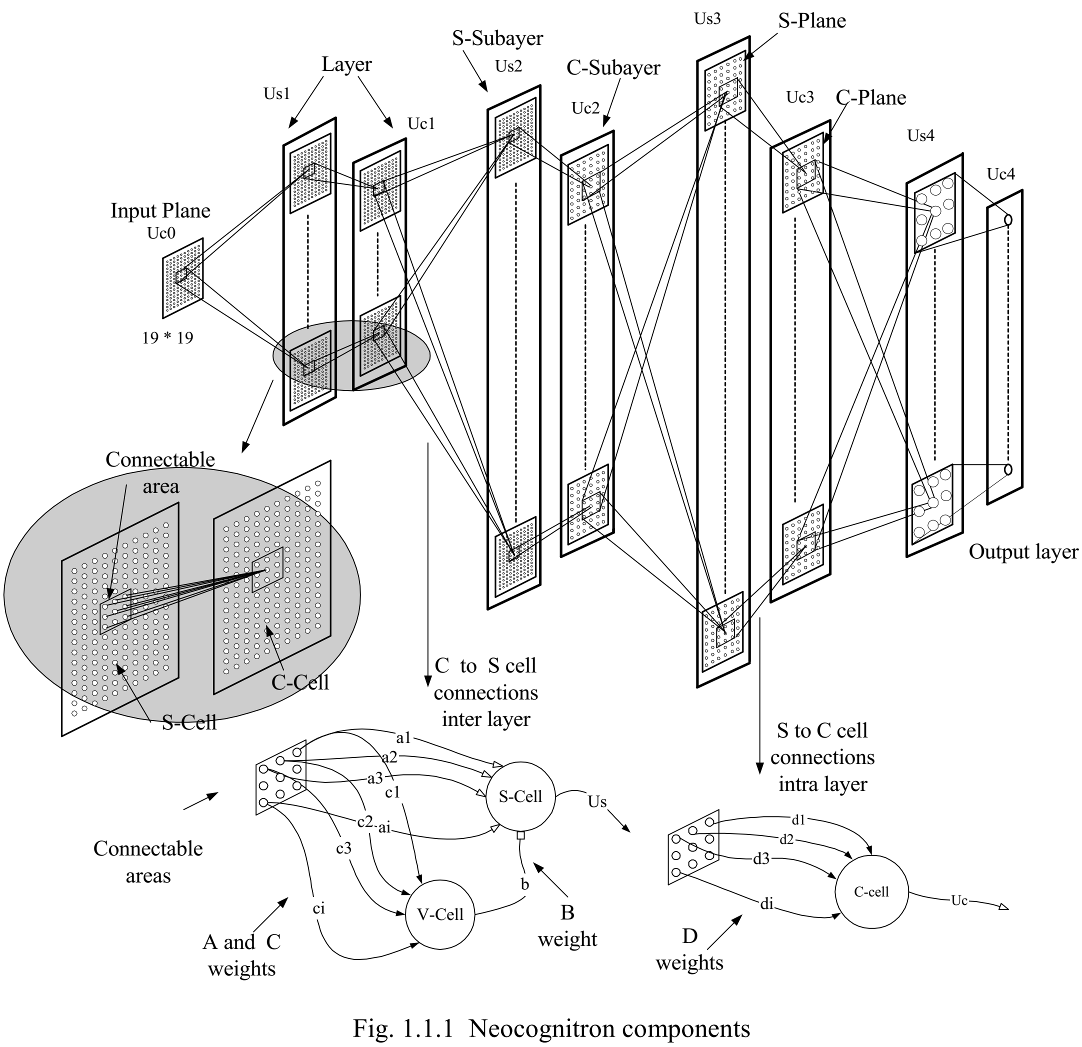

Fig. 1.1.1 Neocognitron components

In figure 1.1.1, a structure model of the neocognitron is depicted. A neocognitron has a cascaded layer structure. The layers contain sublayers, the sublayers on their turn contain different type of planes. Finally planes are build out of cells. There are defined three types of cells. In this chapter the specification of the neocognitron simulator is given. The simulator has been build in software according to the model given in figure 1.1.1.

The Neocognitron simulator has been build in one executable file but consists of two major logical modules: the dialog and the network. The dialog or userinterface is constructed using MFC (MicroSoft Foundation Classes), the neocognitron network is coded in ANSI C++ and is available as a static windows library. The code used to build the neocognitron library was ported from a SUN C++ implementation created by Michal Steuer. The basic functionality of the program has not been changed, only the internal structure of the software has been changed. Classes have been renamed, methods have been renamed or have been moved from one class to another, some classes were added, operations have been rewritten etc. The original source code created by Michal Steuer has been refactored in order to let it run on a PC. As with the original Unix implementation, the userinterface and the neocognitron functions are independent from each other. From the userinterface only certain neocognitron network functions are available all implementation details of the neocognitron are hidded. Besides the neocognitron library `(NeoOv.Lib)` a utility library is available which provides for logging and debugging functions. This library is used by both the `NeoOV.Lib` and the `NeoWin.Exe`.
Currently the network configuration and training set data can only be input from file. The format of the file to be used is specified in section 1.7 of this appendix. Retrieving data from the neocognitron can be performed online from the userinterface. The relative excitation values of the cells in the hidden layers are shown on the graphical main window. This window also shows the absolute excitation values of the output neurons in the last layer of the network.

The software architecture allows a trained neocognitron network to be used in any software system that needs a recognition mechanism for images. The `NeoOv.Lib` can simply be linked into a user application. At runtime you simple `Install` a trained network from a input file and call the recognition method `Test` to recognize image samples supplied in your application. Below a pseudo-code fragment is given to include a network in an application.

```
#include "network.h"

float* z;              // pointer to array of all networks output cells
int    max_layer;      // integer defining the layer number

mNetWork = new CNetwork();                              // the network object
pOutput  = new float[what ever number of output neurons] // output array

// load the network configuration
mNetWork->Install(SupervisedFukushima, a file cont. the netw.configuration);


// let the network recognise and input sample
mNetWork->Test(a file containing the image sample);      // or the next line
mNetWork->Test(a pointer to an array containing the image sample);

// get the networks output
z = mNetWork->GetPlaneOutput(maxlayer, 'C');
memcpy(pOutput,z, what ever number of output neurons);
```

1.2 Neocognitron OO implementation

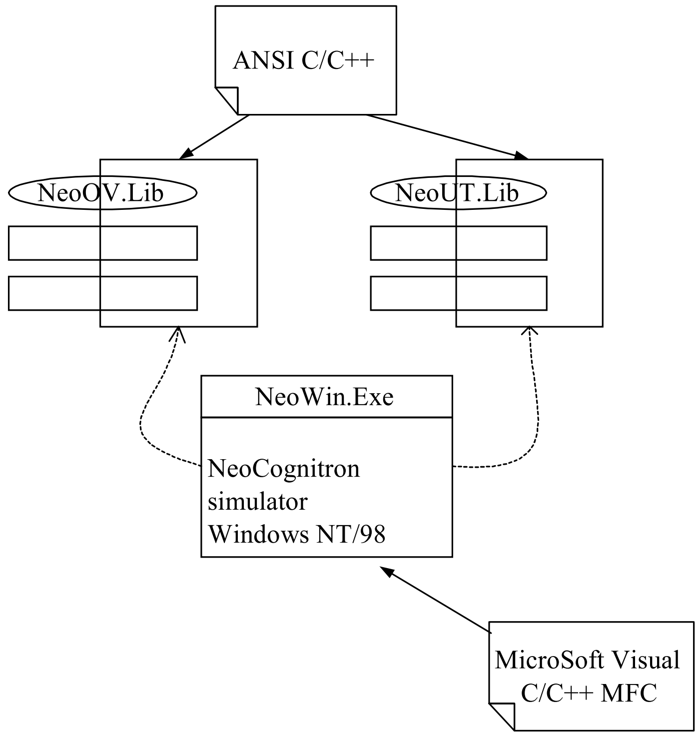

Fig. 1.1.2 The neocognitron simulator software structure

In figure 1.1.2 the basic structure of a neocognitron simulator software is given. Both the network library and the utility library software as well as the userinterface are modelled in C++ Classes.
The root of the implementation is formed by an object of the network class `CNetwork`. A network object is associated with one more objects from a `Clayer` class. Layer objects on their turn are associated with a series of planes from a base `CPlane` class. Different types of planes (S-,V- or C-type) planes are derived from the base CPlane class into the specialized class `CplaneV`, `CplaneC` and `CplaneS`.
Finally weight objects of `CweightA, CweightB, CweightC` or `CweightD` are connected to applicable planes objects. In section 1.4 a complete class diagram of the neocognitron library is given.

1.3 Neocognitron simulator userinterface

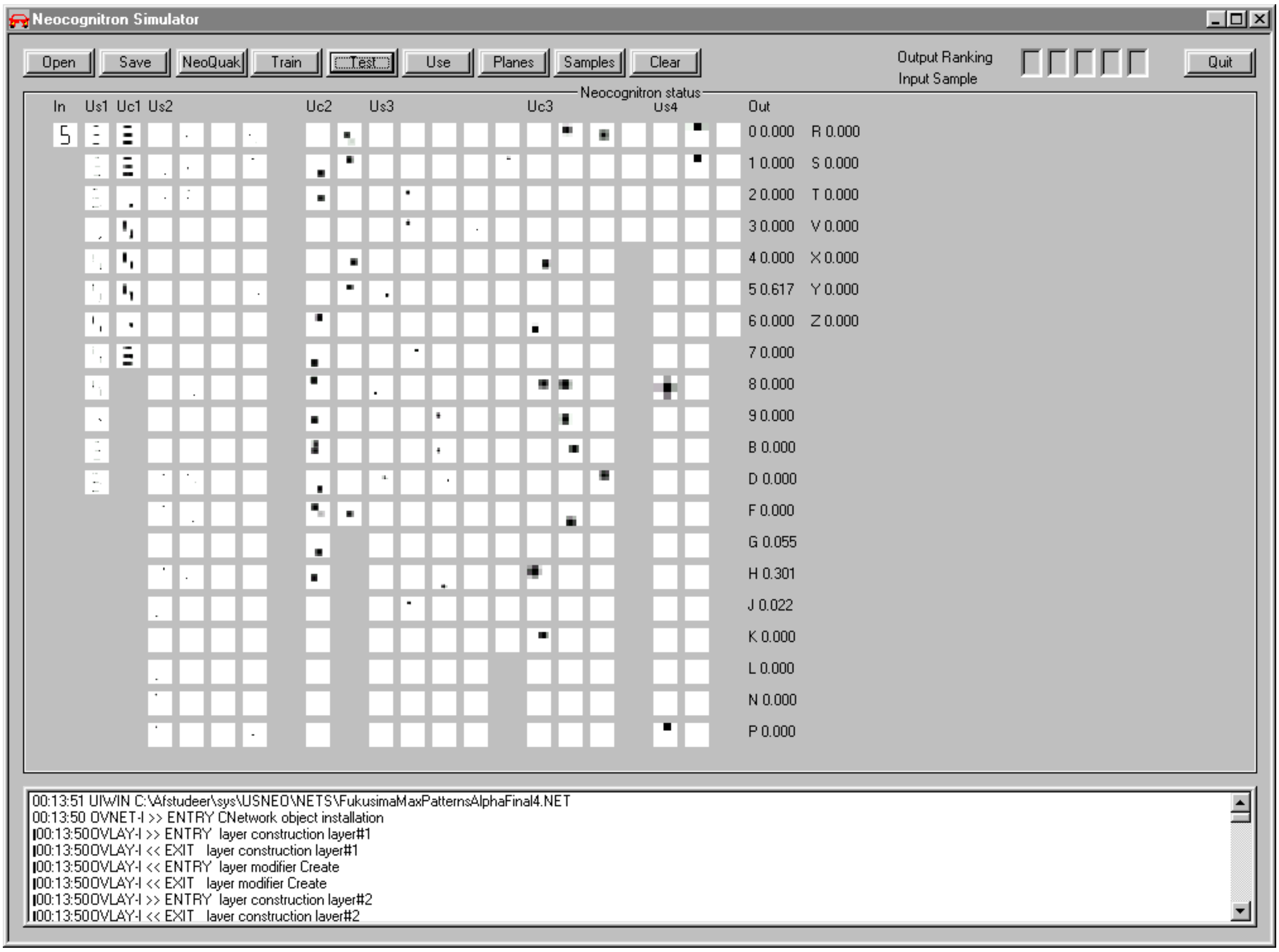

Fig. 1.1.3 The Neocognitron userinterface

Since there is not a million things you can do with a neocognitron, the userinterface of the simulator is kept simple. Figure 1.1.3 shows an example of the response of a 4-layer network configured to classify 27 different characters. The patterns printed in the white squares represent the relative excitation values of individual cells in a plane. Black areas indicate high excitation values; white areas indicate cell excitation values of 0. Different gray values indicate excitation values between 0 and the maximum excitation values.
The userinterface has been build in one `CDialog` class and provides for the following functions available from the buttons on the top of the window:

Function	Description
Open	Provides a filedialog box to select and load a network file. The network file may either contain data of a new network layout including trainingsset samples, or data of an already trained network. When opening a new network file the userinterface starts a supervised training session automatically.
Save	Provides a filedialog box to select a file, in which a trained network is to be saved.
NeoQuak	Pushing the NeoQuak button initiates the loaded network to adjust it weights. Neoquaking is mandatory between a supervised and unsupervised training session
Train	The Train button starts an unsupervised training session on the network loaded. Before the unsupervised training starts, a filedialogbox is presented in order to allow the user to selected a file in which the training set samples for unsupervised training reside.
Test	The test button activates the network for recognizing an input pattern. The input patterns are to be supplied form a file. Upon pushing the Test button the user is provide a FileDialogbox to select a file containing a patteren to be recognized. The response of the network will be presented like shown in figure 1.3.1 above.
Use	The Use and the Test button have the same function. The Use button will not show the response of the networks individual planes or cells. The classes belonging to the five most activated output cells of the network are shown in the edit boxes in the top-right corner of the screen.
Planes	Push this button to repaint the screen.
Samples	This function is used to show training patterns from a fixed and predefined format file. Currently the number of trainingset patterns per plane in case of superimposed training is set to a fixed number of 4. For any other value, the simulator needs to be recompiled and linked.
Clear	Optionally this button is used to clear the screen before repainting.
Quit	Quits the simulator.

The listbox at the bottom of the window is used to display the logging and tracing text from both the userinterface and the network.

1.4 Neocognitron class diagram

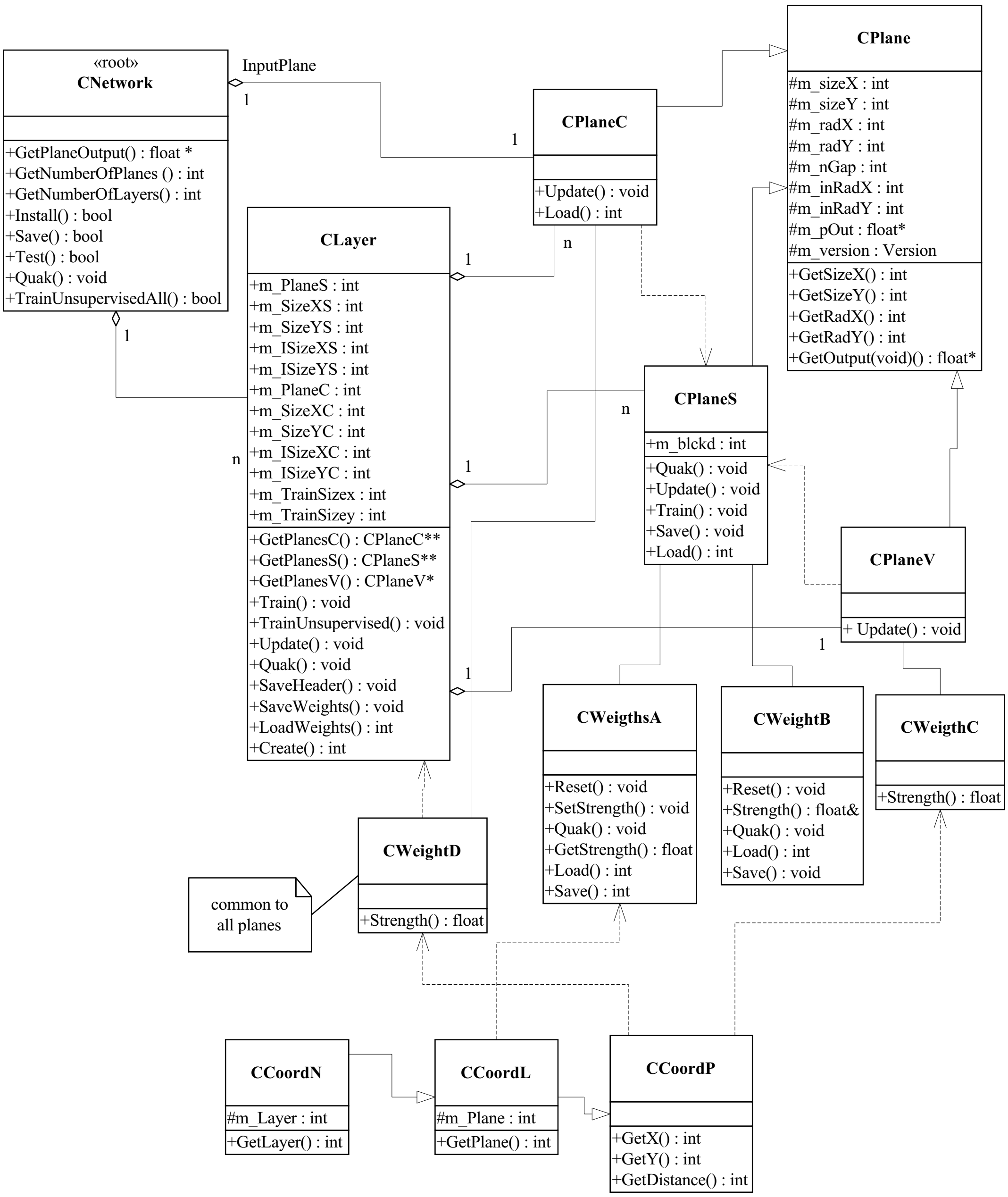

Fig. 1.1.4 The Neocognitron ClassDiagram

A class diagram of the network software is given in figure 1.1.4. The diagram only shows the classes interfaces (public and protected section), private section members are not include in the diagram.

1.5 Neocognitron include dependencies

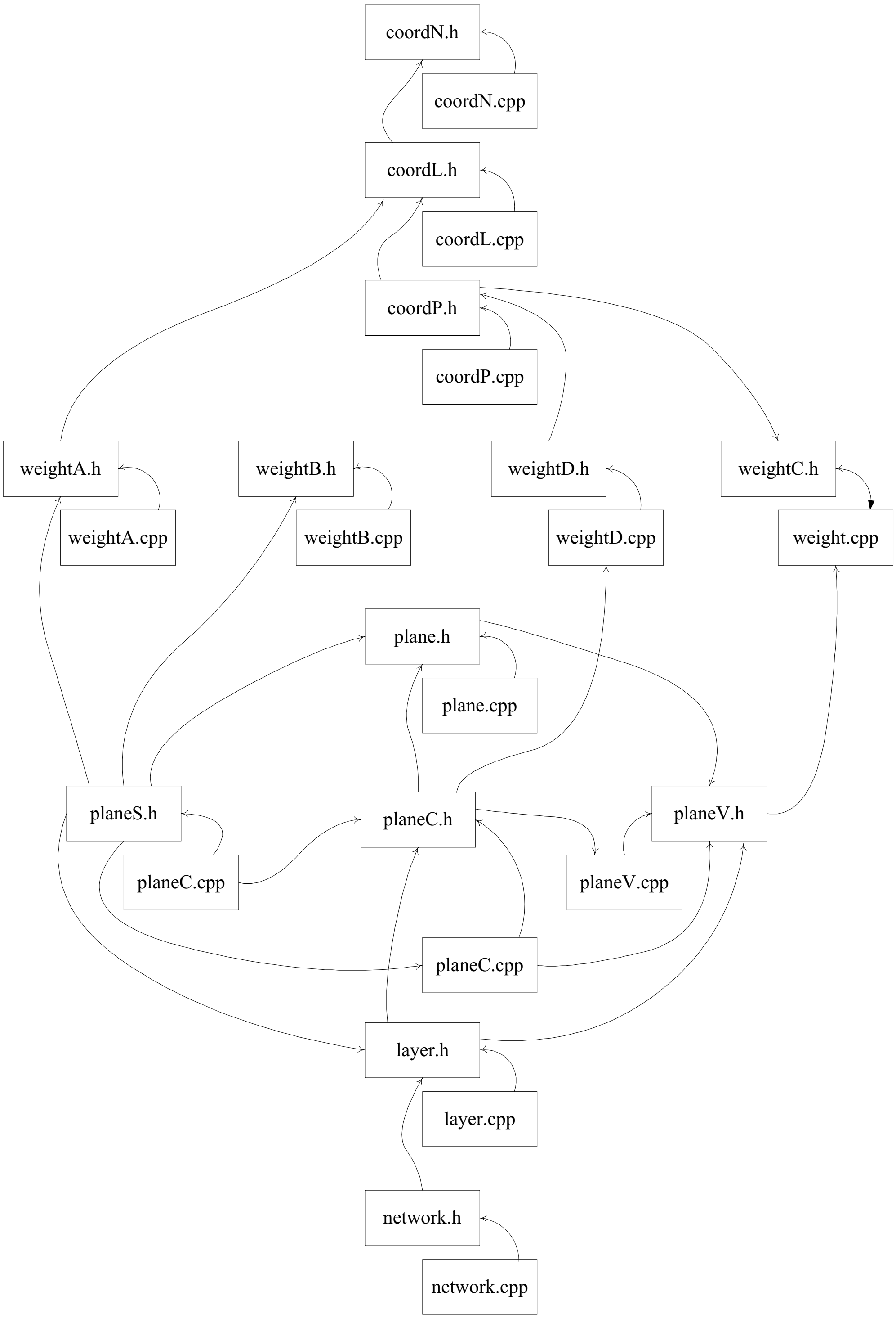

Fig. 1.1.5. The neocognitron includegraph

Figure 1.1.5 shows the dependences between the code files of the neocognitron library software which are relevant when rebuilding the library.

1.6 Neocognitron class descriptions

1.6.1 The network class

CNetwork::CNetwork

<u>Construction</u>

CNetwork( );

Default constructor. Constructs a CNetwork object. The resulting object must be initialized with the **Install** initialization member function.

Return Value
none

Parameters
none

CNetwork::~CNetwork

<u>Destruction</u>

~CNetwork ( );

Call this member function to destruct a network object.

Return Value
none

Parameters
none

CNetwork::GetPlaneOutput

<u>Selection</u>

float* GetPlaneOutput(int layer_number , char layer_type ,int plane_number , int &x, int &y);

Call this function to get the size and exitation of any plane in the network. The method is used to display the plane-pictures on the user interface.

Return Value
Returns a pointer to an array of floating point numbers representing the neuron exictations of the requested plane.

Parameters

layer_number	Specifies the layer sequence number.
Layer_type	Specifies the plane type. Layer_type can have one of the following values: 'C' to retrieve data form a C-type sublayer plane. 'S' to retrieve data from a S-type sublayer plane. 'V' to retreive data from a V-type plane in the sublayer. 'I' to retreive data from the Inputplane of the network.
plane_number	Specifies the plane sequence number.
x	Holds a reference to the size of the plane in horizontal direction
y	Holds a reference to the size of the plane in vertical direction

Remarks
The size of the array returned equals x * y. Neuron exitation values are ordered row by row in the array; starting with the lower row. The network processing layer numbering starts at 0. Planes numbering starts at 0 also. When calling this function with layer_type parameter assigned a 'I' the layer_number and plane_number parameter are ignored. When using this function,

CNetwork::GetNumberOfPlanes and **CNetwork::GetNumberofLayers** should be called first in order to determine the max size of respectively the sublayer and network size. Calling this function with invalid *layer_number* or *plane_number* values will cause most likely an access violation error, so beware !

CNetwork::GetNumberOfPlanes

<u>Selection</u>

int GetNumberOfPlanes (int layer_number , char layer_type);

Call this member function to retreive the number of planes of specified type in a specific layer of the network.

Return Value
Returns the number of planes in the requested sublayer. 0 if the function fails.

Parameters
layer_number Specifies the layer sequence number.

Layer_type Specifies the plane type. Layer_type can have one of the following values:
'C' to retrieve data form a C-type sublayer plane.
'S' to retrieve data from a S-type sublayer plane.

CNetwork::GetNumberOfLayers

<u>Selection</u>

int GetNumberOfLayers (void);

Call this member function to retreive the number of layers in the network.

Return Value
Returns the number of layers in the network.

Parameters
none

CNetwork::Install

<u>Installation</u>

bool Install (Version ver,char *fname);

Call this function to install a network on an already created *CNetwork* object.

Return Value
Returns **true** if succesfully ended, **false** is case of failure.

Parameters

ver Specifies the network version status. The network can be used in the following to states: *SupervisedFukushima* or *UnsupervisedFukushima*
fname A null terminating character string specifing the filename including its path in which the NETWORK and/or TRAINING data resides.

Remarks
This function either installs a 'new untrained' network and succesively trains it supervised or installs an already trained network ready for use. Installation of the netwerk goes in two steps:

First, the network objects layermembers are initialized according to the specifications in the input file. These specifications are: 1) The number of layers, number of planes per layer, size of input layer, structure of each C-type- and S-type sublayer plane size, connectable area, number of planes, nueron gap and s-column. 2) Layer specifications like; selectivity, gamma, delta and deltaneg as well as interconnection specifications betweern S and C layers.
Secondly, either this **Install** methode continues to train the neocognitron supervised by consequetively reading the training samples that follow the network specifications in case the input

file contains the TRAINING keyword or this **Install** methods starts loading the network weights factors and selectivity per plane in case the input files contains the keyword NETWORK.

Failure of this method most likely occure upon invalid input file formats or memory allocations failures in case of hugh a network. Refer to the input file format specification and input file samples in section 1.7.

CNetwork::Save

<u>Installation</u>

bool Save (char *fname);

Call this function to save a trained network on disk.

Return Value
Returns **true** if succesfully ended, **false** is case of failure.

Parameters

fname	A null terminating character string specifing the filename including its path in which the weight data of CNetwork object is to be stored.

Remarks
Saving the network goes in two steps:

First, the network objects specification parameters are saved in a the so-called network HEADER. The HEADER contains data about: the number of layers, number of planes per layer, size of input layer, structure of each C-type- and S-type sublayer: plane size, connectable area, number of planes, nueron gap and s-column. Layer specs like; selectivity, gama, delta and deltaneg as well as interconnection specifications betweern S and C layers.
Secondly, the weight factors per plane as well as the selectivity factor are written to disk.
Failure of this method most likely occure upon write errors to the output file.

CNetwork::Test

<u>Selection</u>

bool Test (char *fname, int savelog=0);

Call this function to recognize an input sample.

Return Value
Returns **true** if succesfully ended, **false** is case of failure.

Parameters

Fname	A null terminating character string specifing the filename including its path in which the input sample is stored.
savelog	Optional parameter to specifiy whether the exitation values of the output neurons are the be save in a logfile. Currently the logfilename cannot be specified but is rather hard coded to *c:\\log.txt*. if *savelog* equals 1 the output data is written to the logfile.

Remarks
This method makes the network operate in a mode it is intended for; recognizing input pictures. When this method is called the input file data is written and stored into the input plane nuerons, consecutively the input values are propagated thru the networks sublayers.
Read the output logfile to retreive the networks response to the input or call the **CNetwork::GetPlaneOutput** methode to get this data. Failure of this function most likely are caused by file i/o problems.

CNetwork::Recl

<u>Modification</u>

void CNetwork::Recl(int till);

Call this function to propagate an input pattern through the network.

Return Value

None.

Parameters
till An integer specifying the layer number untill the input sample number is to be propagated.

CNetwork::Recognise

Selection

char *Recognise (unsigned char sample*);

Call this function to recognize an input sample.

Return Value
Returns a null terminated 5 character string. This string contains the character values belonging to the five most highest character classes the input sample belongs to.

Parameters
sample null terminating character string specifing the input sample.

Remarks
This methode makes the network operate in a mode it is intended for; recoginizing input pictures. When this method is called the input image data is written and stored into the input plane nuerons, consecutively the input values are propagated thru the networks sublayers.
Read the output logfile to retreive the networks response to the input or call the **CNetwork::GetPlaneOutput** methode to get this data. Failure of this function are most likely caused by file i/o problems.

CNetwork::Quak

Selection

void Quak (void);

Call this function to adjust weights and tresholds before a unsupervised training session is started.

Return Value
none

Parameters
none

Remarks
Quak sets the all planes in the blocked state. Calling the methode **CNetwork::TrainUnsupervisedAll** succesively deblocks the networks planes. The Quak function on the **CNetwork** object level calls succesively the Quak method for each plane in the network.

CNetwork::TrainUnsupervisedAll

Selection

bool TrainUnsupervisedAll (char *fname, int strategy=random);

Call this function to perfrom an unsupervised training session on the neocognitron network.

Return Value
Returns **true** if succesfully ended, **false** is case of failure.

Parameters
fname A null terminating character string specifing the filename including its path in which the training sample filenames are stored.
Strategy Optional parameter to specifiy the selection of training files. This parameter may have either one of the following values:

Random	When selected random mode the unsupervised traing procedure randomly selects a file form all avialable ones specified in the input file.
Systematic	When selected systematic the training procedure selects the training sample files in the order the appear in the input file.

Remarks
Failure of this function is most likely caused by file i/o operation errors.

1.6.2 The layer class

Public data Members CLayer objects

m_PlaneS	Integer number of S-planes in the layer
m_SizeXS	Integer number: plane size in horizontal direction in the S-sublayer
m_SizeYS	Integer number: plane size in vertical direction in the S-sublayer
m_ISizeXS	connectable area size in the S-sublayer
m_PlaneC	Integer number of S-planes in the layer
m_SizeXC	Integer number: plane size in horizontal direction in the S-sublayer
m_SizeYC	Integer number: plane size in vertical direction in the C-sublayer
m_ISizeXC	connectable area size in the C-sublayer
m_TrainSizex	X-size of training pattern
m_TrainSizey	Y-size of training pattern

CLayer::CLayer

Construction

CLayer(Version ver,int iprevPlanes,int iLayerID,CPlaneC *iplaneI);

Constructs a CLayer object. The resulting object will be initialized however call the **Clayer::Create** initialization member function to allocate and assign the associated sublayer network planes.

Return Value
none

Parameters

ver	Specifies the network version status. The network can be used in the following to states: *SupervisedFukushima* or *UnsupervisedFukushima.*
IprevPlanes	Integer number specifying the number of previous Cplanes connected.
IlayerID	Integer number specifying a layerindentification.
IplaneI	A pointer to a **CPlaneC** object used to save training patterns on disk during self-organizing.

CLayer::~ CLayer

Destruction

~ CLayer ( );

Call this member function to destruct a layer object.

Return Value
none

Parameters
none

Remarks
It is still to be verified that deleting a layer also frees all associated plane objects. Be prepared for memory leaks.

CLayer::GetPlanesC

Selection

CPlaneC **GetPlanesC();

Call this member function to get a pointer to an array to the **CplaneC** objects associated with this layer.

Return Value

CPlaneC **

Parameters
none

CLayer::GetPlanesS

<u>Selection</u>

CPlaneS **GetPlanesS();

Call this member function to get a pointer to an array to the **CplaneS** objects associated with this layer.

Return Value
CPlaneS **

Parameters
none

CLayer::GetPlanesV

<u>Selection</u>

CPlaneV *GetPlanesV();

Call this member function to get a pointer to the **CplaneV** object associated with this layer.

Return Value
CPlaneV *

Parameters
none

Remarks
Please note a network layer shares only one **CplaneV** object.

CLayer::Greate

<u>Modification</u>

int Create (FILE *iFile , CPlaneC **iPreviousPlaneC,
int iNumPreviousPlanes);

Call this member function to install a network layer according to the input file specifications.

Return Value
An integer indication succes or failure; 1 indicates succes 0 indicates failure.

Parameters

iFile	Specifies a filepointer. This filehandle points to a file containing the networks parameters.
iPreviousPlaneC	Specifies an pointer to a **CplaneC** object array of the previuos layer.
INumPreviousPlanes	Integer number specifying the number of **CplaneC** objects in the previous layer.

Remarks
This function assigns the objects public datamembers values as specifid in the TRAINING or NETWORK file, allocates memory for individual planes and establishes inter (sub)layer connections within the layer object. Only failures of file i/o operations are reported.

CLayer::Train

<u>Modification</u>

void CLayer::Train(int iplane, int iX, int iY)

Call this member function to train a network supervised.

Return Value
None.

Parameters

iPlane	Specifies the **CplaneS** object sequence number within the layer to be trained.
iX	Specifies the size of the training sample in the X-direction.
iY	Specifies the size of the training sample in the Y-direction.

Remarks
Note that training a neocognitrion only applies to a Plane of S-types. This is because only S-type neurons are linked by modifiable input-weights. V-type and C-type neuron do not have modifiable weights.

CLayer::TrainUnsupervised

<u>Modification</u>

void TrainUnsupervised( );

Call this member function to train a network in a Unsupervised mode.

Return Value
None.

Parameters
none

Remarks
Rather complex implementation !!

CLayer::Update

<u>Modification</u>

void Update ( );

Call this member function to recalculate the neuron exitations of both C- and S plane types in both sublayers of the layer under consideration.

Return Value
none.

Parameters
none

CLayer::Quak

<u>Modification</u>

void Quak (void);

Call this function to adjust weight and tresholds of all **CplaneS** objects belonging to the layer.

Return Value
none

Parameters
none

CLayer::SaveHeader

<u>Input/Output</u>

void SaveHeader (FILE *out);

Call this member function to save the Header data of the layer of a trained network.

Return Value
none.

Parameters
Out Specificies a file pointer to the outputfile.

CLayer::SaveWeights

<u>Input/Output</u>

void SaveWeights (FILE *oFile);

Call this member function to save the weight factors associated to CplaneS objects of the sub-layer of the trained network.

Return Value
none.

Parameters
oFile Specificies a file pointer to the outputfile.

Remarks
Normally the outputfile parameter will have the same value as the output parameter with which the *Clayer::SaveHeader* is called. The network header data and the network weight values are stored in the same file.

CLayer::LoadWeights

<u>Input/Output</u>

int LoadWeights(FILE *iFile);

Call this member function to load the weight factors associated to CplaneS objects of the sub-layer of the trained network.

Return Value
An integer indication succes or failure; 1 indicates succes 0 indicates failure.

Parameters
iFile Specificies a file file pointer to the inputfile containing weigths of type A, B and the selectivity parameters.

Remarks
Only failures of file i/o operations are reported.

1.6.3 The plane classes

The CPlane class provides the base functionality of all Plane objects of either V-,S- or C-type. The **CPlane** objects holds basic data member attributes common to all Plane objects.

Protected data Members CPlane objects:

m_ sizeX	Integer number: plane size in horizontal direction
m_ sizeY	Integer number: plane size in vertical direction
m_ radX	Integer number: plane radius in horizontal direction
m_ radY	Integer number: plane radius in horizontal direction
m_ nGap	Integer number: specifiing the neuron gap
m_ inRadX	Integer number: radius of input connections in x-direction
m_ inRadY	Integer number: radius of input connections in y-direction
m_ pOut	Pointer to an array of floating point numbers representing the plane neuron output exitations
m_ version	Enum type specifying type of network. SupervisedFukushima or UnsupervisedFukushima

CPlane::CPlane

<u>Constructor</u>

CPlane(Version ver,int iRadX, int iRadY, int iInRadX,int iInRad,
int inGap);

Call this member function to construct a base **CPlane** object.

Return Value
none

Parameters

ver	Specifies the network version status. The network can be used in the following to states:*SupervisedFukushima* or *UnsupervisedFukushima*
iRadx	Specifies the radius of the plane in the x-direction.
iRady	Specifies the radius of the plane in the y-direction.
iInRadx	Specifies radius of input connections in x-direction.
iInRadx	Specifies radius of input connections in y-direction.
inGap	Integer number: specifiing the neuron gap

Remarks
During construction the neuron excitations of the plane are intialised to 0. The radius of a plane is defined as sixe/2. The implementation restricts only odd values of size in either directions are allowed.

CPlane::~ CPlane

<u>Destruction</u>

~ CPlane ( );

Call this member function to destruct a base plane object.

Return Value
none

Parameters
none

Remarks
Only the planes common attribute m_p0ut is deleted.

CPlane::GetSizeX/Y

Selectors

Int GetSizeX/Y( );

Call this member function to retreive the plane size in x/y-direction.

Return Value
The value of m_ sizeX/Y data attribute.

Parameters
none

Remarks
Note this discription covers for both methodes **CPlane::GetSizeX** and **CPlane::GetSizeY.**

CPlane::GetRadX/Y

Selectors

Int GetSizeX/Y( );

Call this member function to retreive the plane radius in x/y-direction.

Return Value
The value of m_ radX /Y data attribute.

Parameters
none

Remarks
Note this discription covers for both methodes **CPlane::GetSRadX and CPlane::GetRadY.**

CPlane::GetOutput

Selectors

float * GetOutput (void);
float GetOutput (int iX, int iY);

Call this member function to retreive the planes output value(s).

Return Value
Returns a pointer to an array floating point numbers representing the neuron exictations of the requested plane.
Returns a float point number representing the neuron exictation at the (x,y) location on the plane.

Parameters
iX Specifies the neuron location in the X-direction.
iY Specifies the neuron location in the Y-direction.

Remarks
The size of the array returned equals m_sizeX * m_sizeY. Neuron exitation values are ordered row by row in the array, starting with the lower row.

Public data Members CPlaneS objects

m_ blckd	Integer number specifying whether the object is blocked for training.

CPlaneS::CPlaneS

Constructor

CPlaneS(Version ver, int iRadX, int iRadY,

, int iInRadX,int iInRadY,
, int inGap, float iSelectivity,
, int prevPlanes, CWeightsC *iWc);

Call this member function to construct a **CPlaneS** object.

Return Value
none

Parameters

ver	Specifies the network version status. The network can be used in the following to states: *SupervisedFukushima* or *UnsupervisedFukushima.*
iRadx	Specifies the radius of the plane in the x-direction.
iRady	Specifies the radius of the plane in the y-direction.
iInRadx	Specifies radius of input connections in x-direction.
iInRady	Specifies radius of input connections in y-direction.
inGap	Integer number: specifiing the neuron gap
iSelectivity	floating point number specifying the selectivy for the plane
prevPlanes	Number of previuos C-Planes connected to this S-plane type.
iWc	A pointer to a floating point array specifying the WeightC factors for the plane.

Remarks
Reinforcing the b-weights is accomplished within this class, this is the reason the associated c-weights are nessecary as input parameter in the constructor.

CPlaneS::~ CPlaneS

Destruction

~ CPlaneS ( );

Call this member function to destruct a **CplaneS** object.

Return Value
none

Parameters
none

Remarks
When calling this function the associated weightA and WeightB factor space is deallocated.

CPlaneS::Quak

Modification

Quak ( );

Call this member function to "Quake" the **CplaneS** objects weigthA and WeigthB factors.

Return Value
none

Parameters
none

Remarks
When calling this function the m_blckd datamember is set to 1.

CPlaneS::Update

Modification

void Update(CPlaneC **pPlC, int nPlC ,CPlaneV *pPlV);

Call this member function to recalculate the neuron exitations based on current weigth factors of type A en Type B.

Return Value
none

Parameters

pPlc	A pointer to a **CplaneC** object array in the layer under consideration.
NPlc	*Specification of* the number of ***CplaneC*** objects connected to this plane.
PPlV	A pointer to a CplaneV object in the layer under consideration.

CPlaneS::Train

<u>Modification</u>

void Train (CPlaneC **pPl, int nPl, CPlaneV *pPlV, int iX,int iY);
void Train (CPlaneC **pPl, int nPl, CPlaneV *pPlV, int iX,int iY, float value, int unsupervised=1);

Call this member function to recalculate the nueron exitations based on current weigth factors of type A en Type B.

Return Value
none

Parameters

pPlc	A pointer to a **CplaneC** object array in the layer under consideration.
NPlc	Specification of the number of **CplaneC** objects.
PPlV	A pointer to a CplaneV object in the layer under consideration.
iX	Specifies the size of the training sample in X-direction.
iY	Specifies the size of the training sample in Y-direction.
value	Unused.
unsupervised	Optional parameter specifies supervised or unSupervised training, this parameter is used to set the training coefficient internally.

CPlaneS::Save

<u>Input/output</u>

void Save(FILE *oFile);

Call this member function to save the assiociated weightA and weightB factors of the **CplaneS** object to disk.

Return Value
none

Parameters

oFile	Specificies a file file pointer to the outputfile in which weigths of type A, B and the selectivity parameters are to be stored.

CPlaneS::Load

<u>Input/output</u>

int Load(FILE *iFile);

Call this member function to recall the assiociated weightA and weightB factors of the **CplaneS** object from disk.

Return Value
An integer indication succes or failure; 1 indicates succes 0 indicates failure.

Parameters

oFile Specificies a file file pointer to the inputfile in which weigths of type A, B and the selectivity parameters were saved.

Remarks

Only failures of file i/o operations are reported.

CPlaneV::CPlaneV

Constructor

CPlaneV (Version ver,int iRadX, int iRadY,
int iInRadX, int iInRadY,
int inGap, CWeightsC *iwc);

Call this member function to construct a **CPlaneV** object.

Return Value

none

Parameters

ver	Specifies the network version status. The network can be used in the following to states:*SupervisedFukushima* or *UnsupervisedFukushima*
iRadx	Specifies the radius of the plane in the x-direction.
iRady	Specifies the radius of the plane in the y-direction.
iInRadx	Specifies radius of input connections in x-direction.
iInRadx	Specifies radius of input connections in y-direction.
inGap	Integer number specifiing the neuron gap.
iWc	A pointer to a floating point array specifying the WeightC factors for the plane.

CPlaneV::~ CPlaneV

Destruction

~ CPlaneS ( );

Call this member function to destruct a **CplaneV** object.

Return Value

none

Parameters

none

Remarks

When calling this function the associated weightC are not deallocated for memory space.

CPlaneV::Update

Modification

void Update(CPlaneC **pPl,int nPl);

Call this member function to recalculate the neuron exitations based on current weigth factors of type C.

Return Value

none

Parameters

pPl	A pointer to a **CplaneC** object array in the layer under consideration.
NPlc	Specification of the number of **CplaneC** objects connected to this plane.

CPlaneC::CPlaneC

<u>Construction</u>

**CPlaneC (Version ver,int iRadX, int iRadY,
int iInRadX, int iInRadY,
int inGap, CWeightsC *iwc);
CPlaneC(Version ver,int iRadX,int iRadY);**

Call this member function to construct a **CPlaneC** object. The second constructor specified may be called to construct a **CplaneC** object as input plane.

Return Value
None

Parameters

Ver	Specifies the network version status. The network can be used in the following to states:*SupervisedFukushima* or *UnsupervisedFukushima.*
iRadx	Specifies the radius of the plane in the x-direction.
iRady	Specifies the radius of the plane in the y-direction.
iInRadx	Specifies radius of input connections in x-direction.
iInRadx	Specifies radius of input connections in y-direction.
inGap	Integer number: specifiing the neuron gap.
iWc	A pointer to a floating point array specifying the WeightC factors for the plane.

CPlaneC::~ CPlaneC

<u>Destruction</u>

~ CPlaneC ( );

Call this member function to destruct a **CplaneC** object.

Return Value
none

Parameters
none

Remarks
When calling this function the associated weightD are not deallocated for memory space.

CPlaneC::Update

<u>Modification</u>

void Update(CPlaneS **pPl, int nPl);

Call this member function to recalculate the neuron exitations based on current weigth factors of type D.

Return Value
none

Parameters

pPl	A pointer to a **CplaneS** object array in the layer under consideration.
NPlc	*Specification of the number of **CplaneS** objects*

CPlaneC::Load

<u>Input/output</u>

int Load(FILE *iFile, int iRadX, int iRadY);

int CPlaneC::Load(unsigned char *in,int radX,int radY);

Call this member function to load a pattern into the input plane of the network. A call to this method is typically followed by CNetwork::Recl() method to propagate the input pattern through the network.

Return Value
An integer indication succes or failure; 1 indicates succes 0 indicates failure.

Parameters

In	Array containing the input pattern to be recognized. The input pattern is stored row by row starting with the lower row.
iFile	Specificies a file file pointer to the inputfile in which the input pattern resides.
iRadx	Specifies the radius of the plane in the x-direction.
iRady	Specifies the radius of the plane in the y-direction.

Remarks
Only failures of file i/o operations are reported.

1.7 Neocognitron input file description

1.7.1 Input file definition

Below input file definitions of both the training file and the network file are given. Both files have the same header. In this header the network configuration is given. A training file holds the training set samples for a network. A network file holds the a-weights and b-weight values for a trained network.

```
 ↵      = newline
 ε      = void
 exp+   = exp|exp exp*
 exp*   = ε|exp exp*
 BOLD   = terminator symbols
 {comments}
 file-type file_name.trn or file_name.txt is used to train the network
           file_name.net is used to load a trained network
```

```
training-file::=INPUT LAYER↵
                integer integer↵ {xSize  ySize }
                layer-specification+
                TRAINING↵
                training-specification+

layer-specification::=LAYER integer↵ {     layer#}
                      slayer-specification↵
                      clayer-specification↵
                      trainingsample-size↵
                      layer-parameters↵
                      layer-intraconnections

slayer-specification  ::=integer integer integer  integer integer integer   integer
                        {plane x-y       connect.area x-y #planes neurongap sColumn}
clayer-specification  ::=integer integer integer  integer integer integer
                        {plane x-y       connect.area x-y #planes neurongap}
trainingsample-size   ::=integer integer
                        {xSize   ySize}
layer-parameters      ::=float float float float
                        {sel. gamma delta deltaneg}
layer-intraconnections::=(s-layerplanes↵ c-layerplanes↵)+

s-layerplanes::=integer * {if plane has over 20 planes start next layer
                           -intraconnections expression on new line            }
c-layerplanes::=integer * {indicates which c-plane is connected to which s-plane}


training-specification::=NEW LAYER↵
                         last-plane-sample | plane-sample+

plane-sample          ::=(integer +↵)+
                         PNCinteger integer | P_Cinteger integer↵
                        {P_C indicates superimposed training sample}
                        {integer indicate the sample location on input plane}
last-plane-sample     ::=(integer +↵)+
                         PNCinteger integer↵
```

```
network-file::=INPUT LAYER↵
               integer integer↵
               layer-specification+
               NETWORK↵
               network-specification+

network-specification::=NEW LAYER↵
                        plane-weights+

plane-weights        ::=PLANE integer↵ {     plane#}
                       (float +↵)+↵    {a-weights matrix per plane}
                        float↵         {single b-weight per plane}
```

1.7.2 Input file example

Below an input file sample used to train the network described in section 2.5.1 of this appendix.

```
INPUT LAYER
19 19
LAYER 1
19 19 03 03 12 01 02
21 21 03 03 12 01
03 03
1.70  0.90  0.90  4.00
00 01 02 03 04 05 06 07 08 09 10 11
00 01 02 03 04 05 06 07 08 09 10 11
LAYER 2
21 21 05 05 08 01 02
13 13 07 07 08 02
09 09
3.2 0.9 0.8 4.0
00 01 02 03 04 05 06 07
00 01 02 03 04 05 06 07
LAYER 3
11 11 11 11 04 01 02
01 01 05 05 02 02
19 19
1.3 0.9 0.7 1.4
00 01 02 03
00 00 01 01
TRAINING
NEW LAYER
0 0 0
1 1 1
0 0 0
PNC2 2
0 0 1
1 1 0
0 0 0
.
.
PNC2 2
1 0 0
0 1 1
0 0 0
PNC2 2
NEW LAYER
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 1 1 1
0 0 0 0 0 1 0 0 0
0 0 0 0 1 0 0 0 0
0 0 0 1 0 0 0 0 0
0 0 0 1 0 0 0 0 0
0 0 0 1 0 0 0 0 0
P_C5 5
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 1 1 1
0 0 0 0 0 1 0 0 0
0 0 0 0 1 0 0 0 0
0 0 0 0 1 0 0 0 0
0 0 0 0 1 0 0 0 0
0 0 0 0 1 0 0 0 0
P_C5 5
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 1 1 1
0 0 0 0 0 1 0 0 0
0 0 0 0 1 0 0 0 0
0 0 0 1 0 0 0 0 0
0 0 1 0 0 0 0 0 0
0 0 1 0 0 0 0 0 0
PNC5 5
```

```
.
.
.
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
1 1 1 1 1 1 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
PNC5 5
NEW LAYER
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 1 1 1 1 1 0 0 0 0 0 0 0
0 0 0 0 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0
0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0
0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0
0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0
0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0
0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0
0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0
0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0
0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0
0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0
0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0
0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0
0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0
0 0 0 0 0 0 1 1 1 1 1 1 1 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
PNC10 10
.
.
.
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 1 1 1 1 1 0 0 0 0 0 0 0 0 0
0 0 0 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0
0 0 0 0 0 1 0 0 0 0 0 1 1 0 0 0 0 0 0
0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0
0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0
0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0
0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0
0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0
0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0
0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0
0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0
0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0
0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0
0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0
0 0 0 0 0 1 1 1 1 1 1 1 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
PNC10 10
```

Appendix II

Neocognitron network configurations

2.1 Fukisuma's network configuration for handwritten character recognition

This network is described in section 2.3 through section 2.5 in the main document.

Editable training set files:

Layer 2: NeoPatsLayer2Fukusima	TextDocument	31KB
Layer 3: NeoPatsLayer3Fukusima	TextDocument	114 KB
Layer 4: NeoPatsLayer4Fukusima	TextDocument	55 KB

Coded Neocognitron training set file:

FukusimaMaxPatternsAlphaFinal1	TextDocument	287 KB

Runtime Neocognitron network files:

FukusimaMaxPatternsAlphaFinal	Net file	1,653 KB[1]
FukusimaMaxPatternsAlphaFinal1	Net file	1,652 KB[2]

[1] This file has been used to verify the original fukisuma's network on printed characters

[2] This file has been used to verify the orignal fukisuma network on his handwritten input sample. It has slighty different weight values due to a minor code change implemented in the neocognitron simulator s/w.

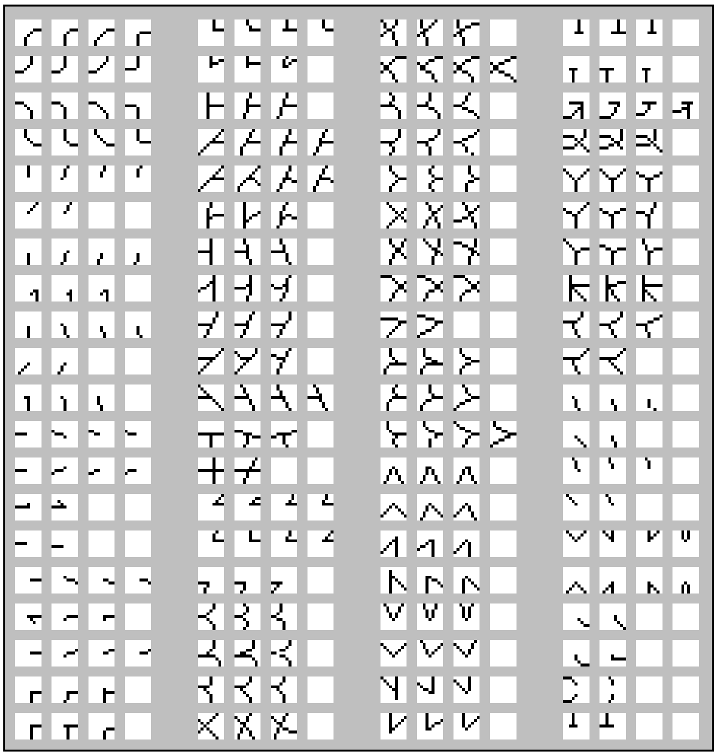

Fig. 2.1.1 Training pattern layer 2 Fukisuma’s orginal network

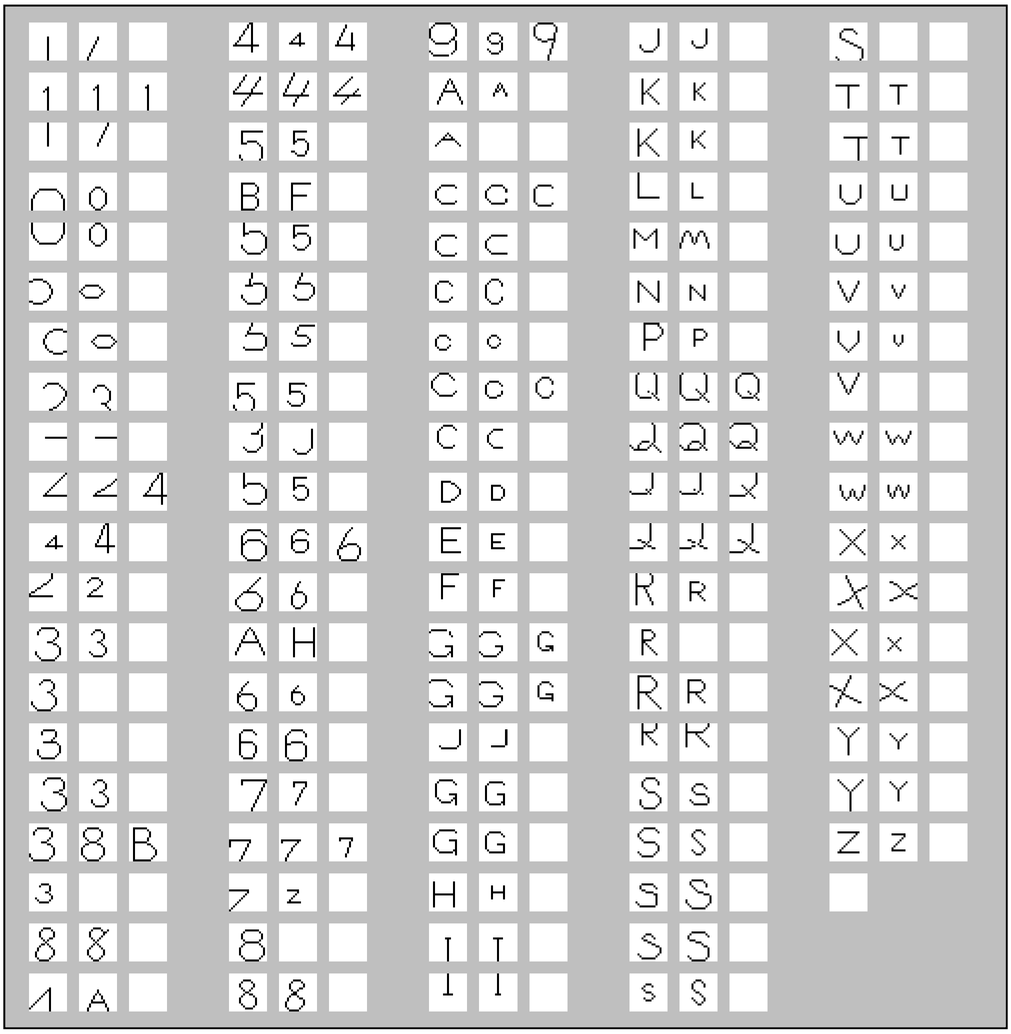

Fig. 2.1.2 Training pattern layer 3 Fukisuma's orginal network

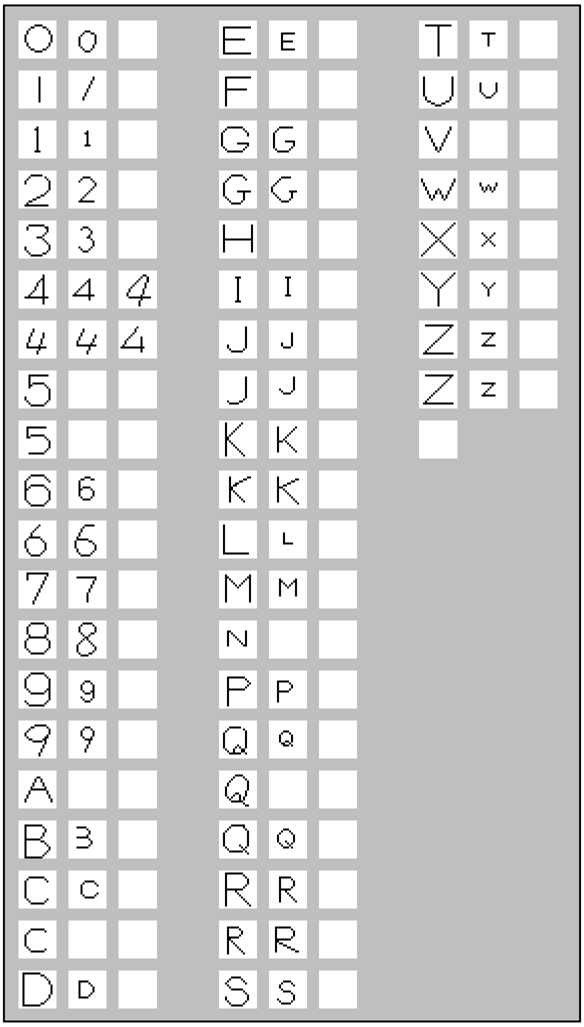

Fig. 2.1.3 Training pattern layer 4 Fukisuma’s orginal network

Table 2.1.1 Fukisuma's orginal network configuration

Layer	Plane size	Connectable Area	Number of Planes	Neuron Gap	Training pattern size
Input	19*19				
S-layer 1	19*19	3*3	12	1	3*3
C-layer 1	21*21	3*3	8	1	-
	Selectivity	**Gamma**	**Delta**	**DeltaBar**	
	1.70	0.90	0.90	4.00	
Sublayer inter connections	0 1 2 3 4 5 6 7 8 9 10 11				
	0 1 1 2 3 3 4 5 5 6 7 7				
S-layer 2	21*21	5*5	80	1	9*9
C-layer 2	13*13	7*7	33	1	-
	Selectivity	**Gamma**	**Delta**	**DeltaBar**	
	4.00	0.90	0.80	4.00	
Sublayer inter connections	0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19				
	0 1 2 3 4 4 5 5 5 5 5 6 6 6 6 7 7 7 8 8				
	20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39				
	9 9 10 10 10 10 11 11 11 11 11 12 13 14 14 15 16 16 16 16				
	40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59				
	16 16 16 16 17 17 17 17 17 17 17 17 18 18 18 18 19 19 19 19				
	60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79				
	20 21 22 23 24 24 24 25 25 25 26 26 27 27 28 29 30 30 31 32				
S-layer 3	13*13	5*5	97	1	19*19
C-layer 3	7*7	5*5	64	2	-
	Selectivity	**Gamma**	**Delta**	**DeltaBar**	
	1.50	0.90	0.70	2.50	
Sublayer inter connections	0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19				
	0 0 1 2 3 4 5 6 6 7 7 8 9 9 9 10 11 11 11 12				
	20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39				
	13 13 14 14 15 15 15 16 17 17 18 18 18 19 19 20 21 21 22 22				
	40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59				
	23 24 24 25 25 26 26 27 27 28 29 30 31 31 32 33 34 35 36 37				
	60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79				
	38 39 39 40 41 42 43 44 44 45 45 46 46 47 48 49 50 50 50 50				
	80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96				
	50 51 52 53 54 55 55 56 57 58 59 59 60 60 61 62 63				
S-layer 4	3*3	5*5	47	1	19*19
Ouput	1*1	3*3	35	1	-
	Selectivity	**Gamma**	**Delta**	**DeltaBar**	
	1.00	0.80	1.0	1.0	
Sublayer inter connections	0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19				
	0 1 1 2 3 4 4 5 5 6 6 7 8 9 9 10 11 12 12 13				
	20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39				
	14 15 16 16 17 18 19 19 20 20 21 22 23 24 25 25 25 26 26 27				
	40 41 42 43 44 45 46				
	28 29 30 31 32 33 34				

2.2 Fukisuma's network configuration for handwritten character recognition using a refined training set

This network is described in section 4.2.1 of the main document.

Editable training set files:

Layer 2: NeoPatsLayer2Fukusima	TextDocument	31KB
Layer 3: NeoPatsLayer3Cornet3	TextDocument	122 KB
Layer 4: NeoPatsLayer4Cornet3	TextDocument	64 KB

Coded Neocognitron training set file:

FukusimaMaxPatternsAlphaNewLayer34_3	TextDocument	287 KB

Runtime Neocognitron network file:

FukusimaMaxPatternsAlphaNewLayer34_3	Net file	1,893 KB

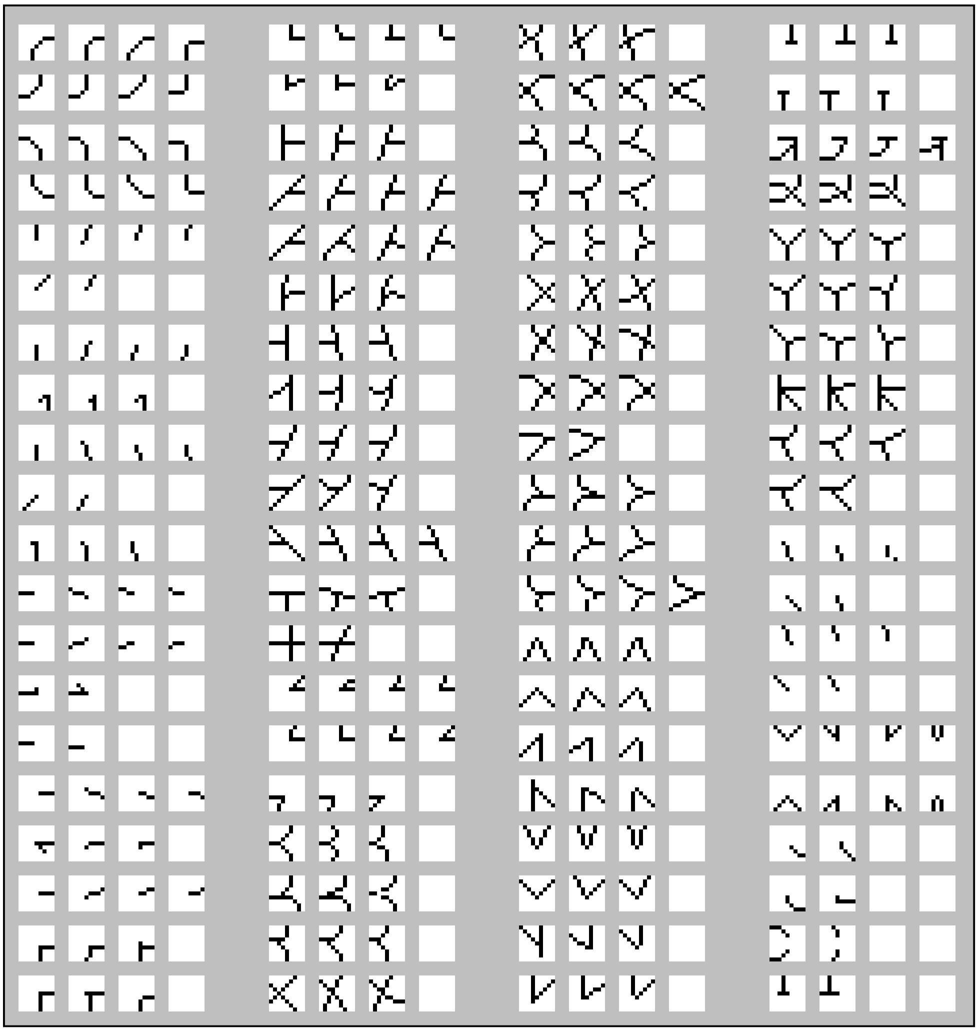

Fig. 2.2.1 Training set pattern layer 2

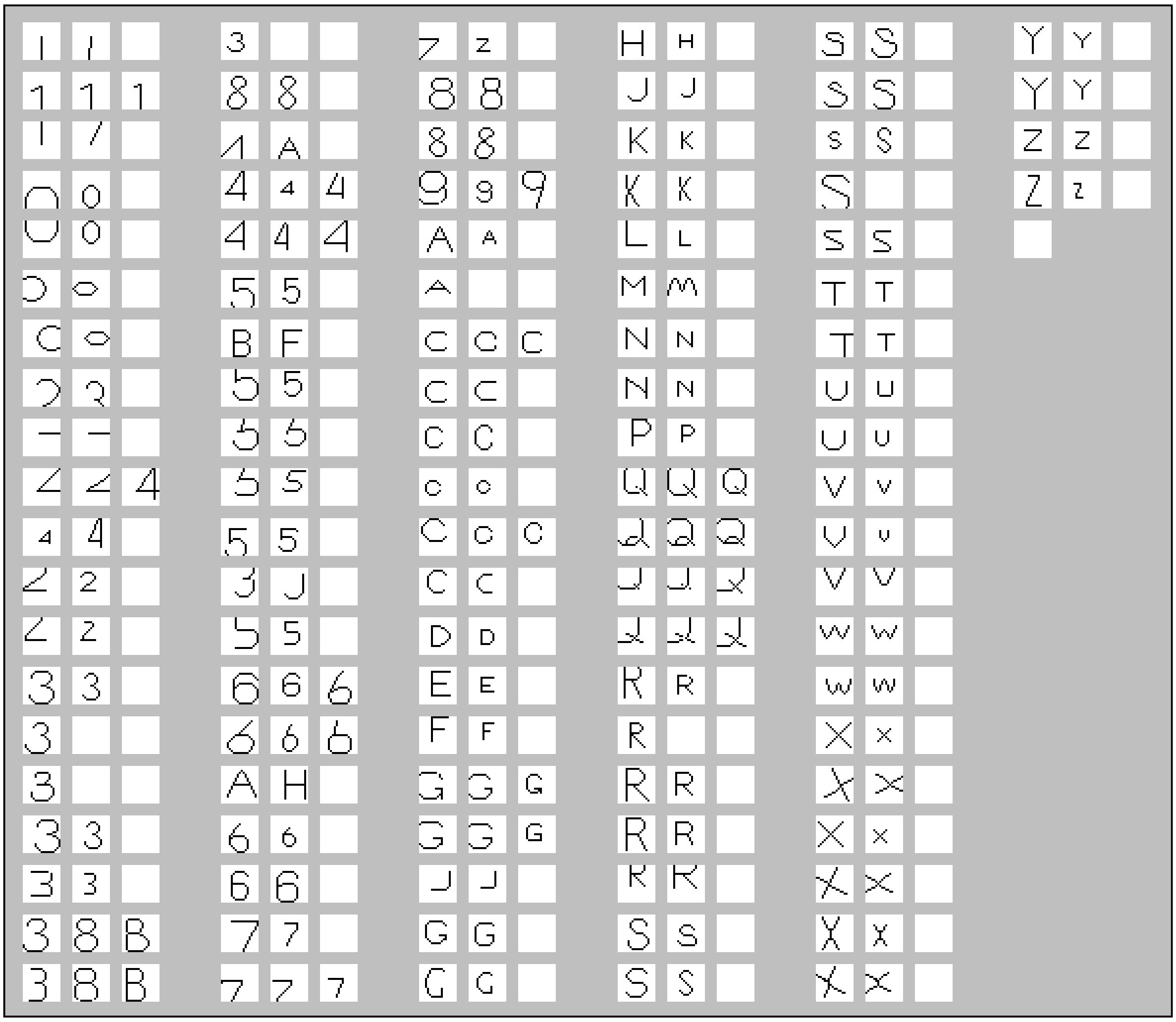

Fig. 2.2.2 Refined training set pattern for layer 3

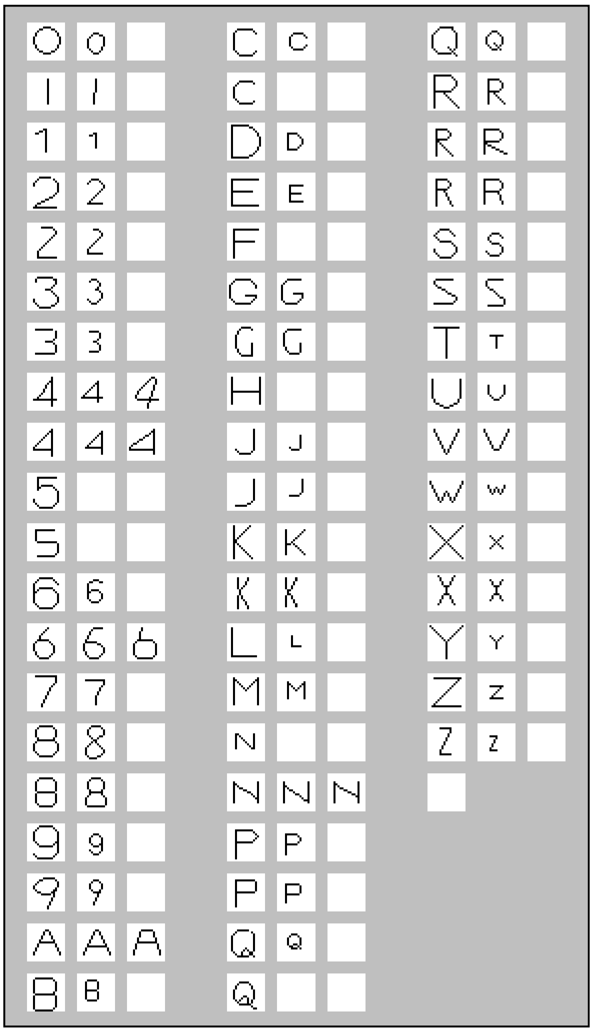

Fig. 2.2.3 Refined training set pattern for layer 4

Table 2.2.1 The refined network configuration

Layer	Plane size	Connectable Area	Number of Planes	Neuron Gap	Training pattern size
Input	19*19				
S-layer 1	19*19	3*3	12	1	3*3
C-layer 1	21*21	3*3	8	1	-
	Selectivity	**Gamma**	**Delta**	**DeltaBar**	
	1.70	0.90	0.90	4.00	
Sublayer inter connections	0 1 2 3 4 5 6 7 8 9 10 11 0 1 1 2 3 3 4 5 5 6 7 7				
S-layer 2	21*21	5*5	80	1	9*9
C-layer 2	13*13	7*7	33	1	-
	Selectivity	**Gamma**	**Delta**	**DeltaBar**	
	3.80	0.90	0.80	4.00	
Sublayer inter connections	0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 0 1 2 3 4 4 5 5 5 5 5 6 6 6 6 7 7 7 8 8 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 9 9 10 10 10 10 11 11 11 11 11 12 13 14 14 15 16 16 16 16 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 16 16 16 16 17 17 17 17 17 17 17 17 18 18 18 18 19 19 19 19 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 20 21 22 23 24 24 24 25 25 25 26 26 27 27 28 29 30 30 31 32				
S-layer 3	13*13	5*5	104	1	19*19
C-layer 3	7*7	5*5	69	2	-
	Selectivity	**Gamma**	**Delta**	**DeltaBar**	
	1.50	0.90	0.70	2.50	
Sublayer inter connections	00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 00 00 01 02 03 04 05 06 06 07 07 08 09 10 10 10 11 12 13 14 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 15 15 16 17 17 18 18 19 19 19 20 21 21 22 22 23 24 24 25 26 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 26 27 27 28 29 29 30 30 31 31 32 32 33 34 35 36 36 37 38 39 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 40 41 42 42 43 44 45 45 46 47 47 48 48 49 49 50 50 51 52 53 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 53 53 53 53 54 55 56 57 57 58 58 59 60 61 62 62 63 63 64 64 100 101 102 103 065 066 067 068				
S-layer 4	3*3	5*5	55	1	19*19
Ouput	1*1	3*3	35	1	-
	Selectivity	**Gamma**	**Delta**	**DeltaBar**	
	1.00	0.80	1.0	1.0	
Sublayer inter connections	00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 00 01 01 02 02 03 03 04 04 05 05 06 06 07 08 08 09 09 10 11 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 12 12 13 14 15 16 16 17 18 18 19 19 20 21 22 22 23 23 24 24 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 24 25 25 25 26 26 27 28 29 30 31 31 32 33 33				

2.3 Fukisuma's network configuration for handwritten character recognition using a modified training set

This network is described in section 4.2.2 of the main document.

Editable training set files:		
Layer 2: NeoPatsLayer2Cornet4	TextDocument	31KB
Layer 3: NeoPatsLayer3Cornet4	TextDocument	101 KB
Layer 4: NeoPatsLayer4Cornet4	TextDocument	57 KB
Coded Neocognitron training set file:		
FukusimaMaxPatternsAlphaNewLayer34_4	TextDocument	252 KB
Runtime Neocognitron network file:		
FukusimaMaxPatternsAlphaNewLayer34_4	Net file	1,884 KB

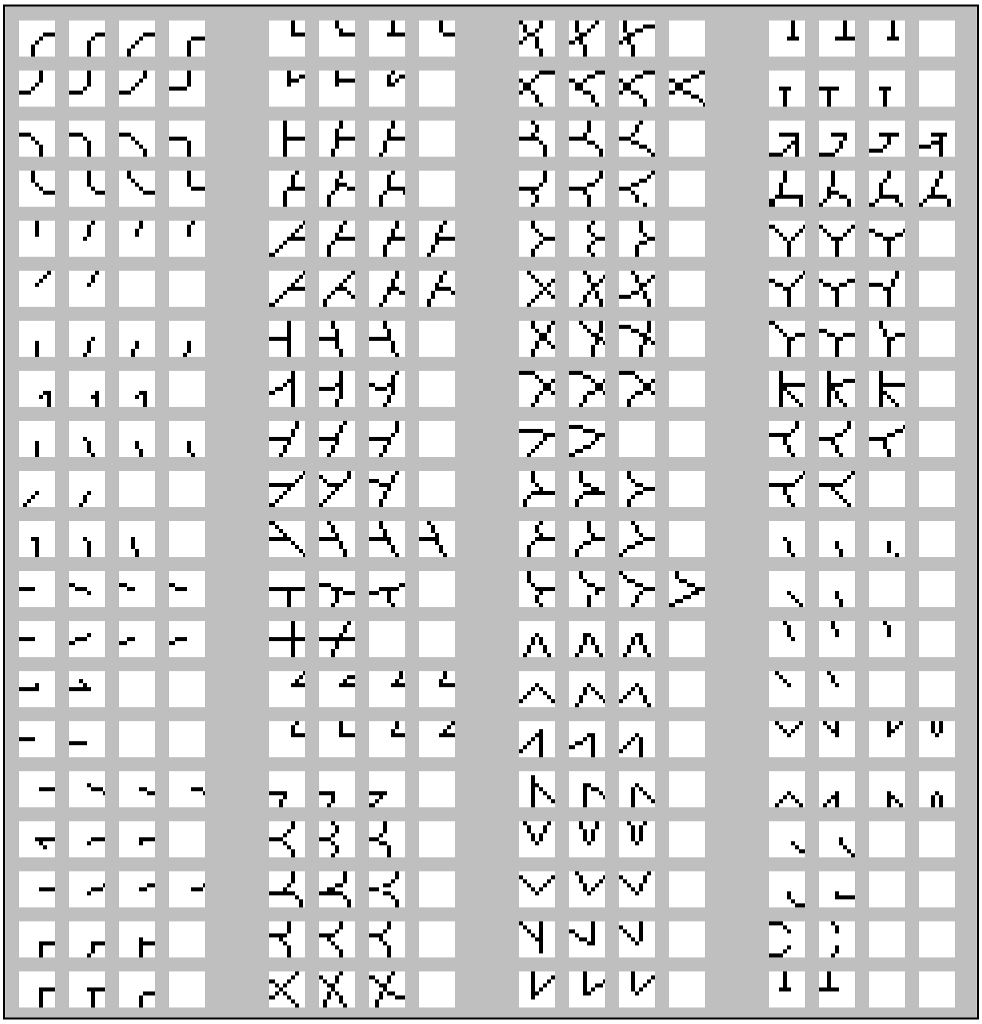

Fig. 2.3.1 Trainingsset patterns layer 2 for license plate characters

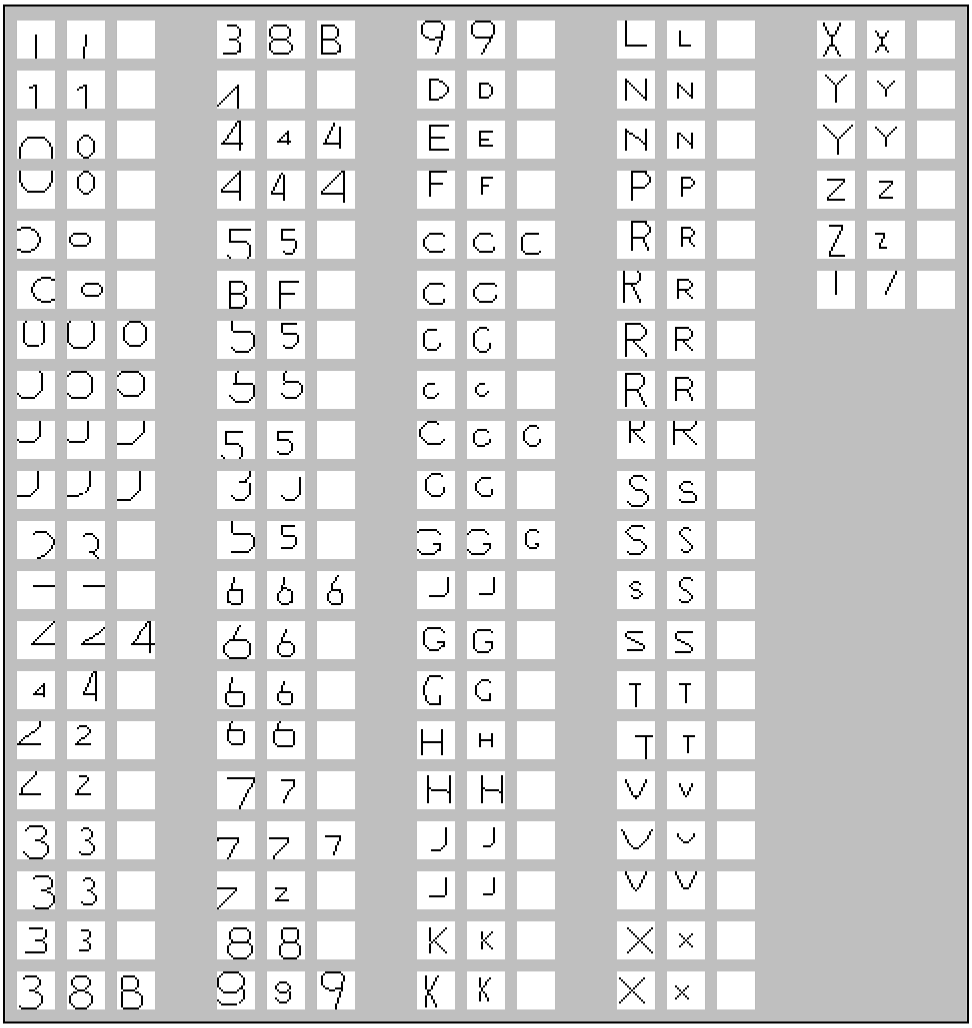

Fig. 2.3.2 Training set patterns layer 3 for license plate characters

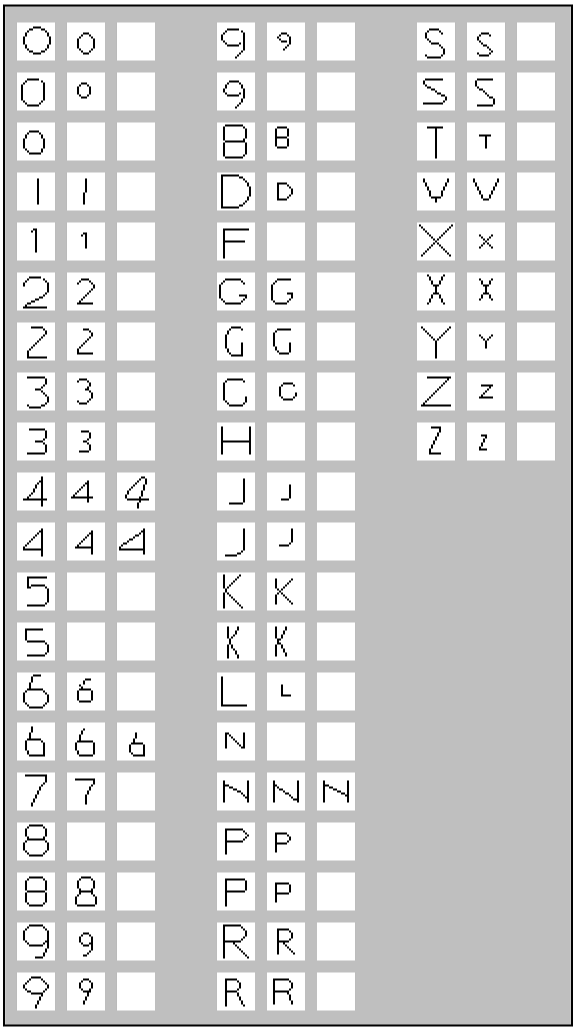

Fig. 2.3.3 Training set patterns layer 4 for license plate characters

Table 2.3.1 The modified network configuration for license plate characters

Layer	Plane size	Connectable Area	Number of Planes	Neuron Gap	Training pattern size
Input	19*19				
S-layer 1	19*19	3*3	12	1	3*3
C-layer 1	21*21	3*3	8	1	-
	Selectivity	**Gamma**	**Delta**	**DeltaBar**	
	1.70	0.90	0.90	4.00	
Sublayer inter connections	0 1 2 3 4 5 6 7 8 9 10 11 0 1 1 2 3 3 4 5 5 6 7 7				
S-layer 2	21*21	5*5	80	1	9*9
C-layer 2	13*13	7*7	33	1	-
	Selectivity	**Gamma**	**Delta**	**DeltaBar**	
	3.80	0.90	0.80	4.00	
Sublayer inter connections	0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 0 1 2 3 4 4 5 5 5 5 5 6 6 6 6 7 7 7 8 8 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 9 9 10 10 10 10 11 11 11 11 11 12 13 14 14 15 16 16 16 16 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 16 16 16 16 17 17 17 17 17 17 17 17 18 18 18 18 19 19 19 19 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 20 21 22 23 24 24 24 25 25 25 26 26 27 27 28 29 30 30 31 32				
S-layer 3	13*13	5*5	86	1	19*19
C-layer 3	7*7	5*5	86	2	-
	Selectivity	**Gamma**	**Delta**	**DeltaBar**	
	1.50	0.90	0.70	2.50	
Sublayer inter connections	00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 80 81 82 83 84 85				
S-layer 4	3*3	5*5	49	1	19*19
Ouput	1*1	3*3	27	1	-
	Selectivity	**Gamma**	**Delta**	**DeltaBar**	
	1.00	0.80	1.0	1.0	
Sublayer inter connections	00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 00 00 00 01 01 02 02 03 03 04 04 05 05 06 06 07 08 08 09 09 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 09 09 10 11 12 13 13 13 14 15 15 16 16 17 18 18 19 19 20 20 40 41 42 43 44 45 46 47 48 21 21 22 23 24 24 25 26 26				

2.4 Three layered network based on thinned character samples of normalised license plate character cut-outs

This network is decribed in section 4.3 of the main document.

Editable training set files:

Layer 2: NeoPatsLayer2Cornet15	TextDocument	33KB
Layer 3: NeoPatsLayer3Cornet15	TextDocument	65 KB

Coded Neocognitron training set file:

meAlphaMaxNewPatternsLayer15	TextDocument	91 KB

Runtime Neocognitron network file:

FukusimaMaxPatternsAlphaLayer15	Net file	2,578 KB

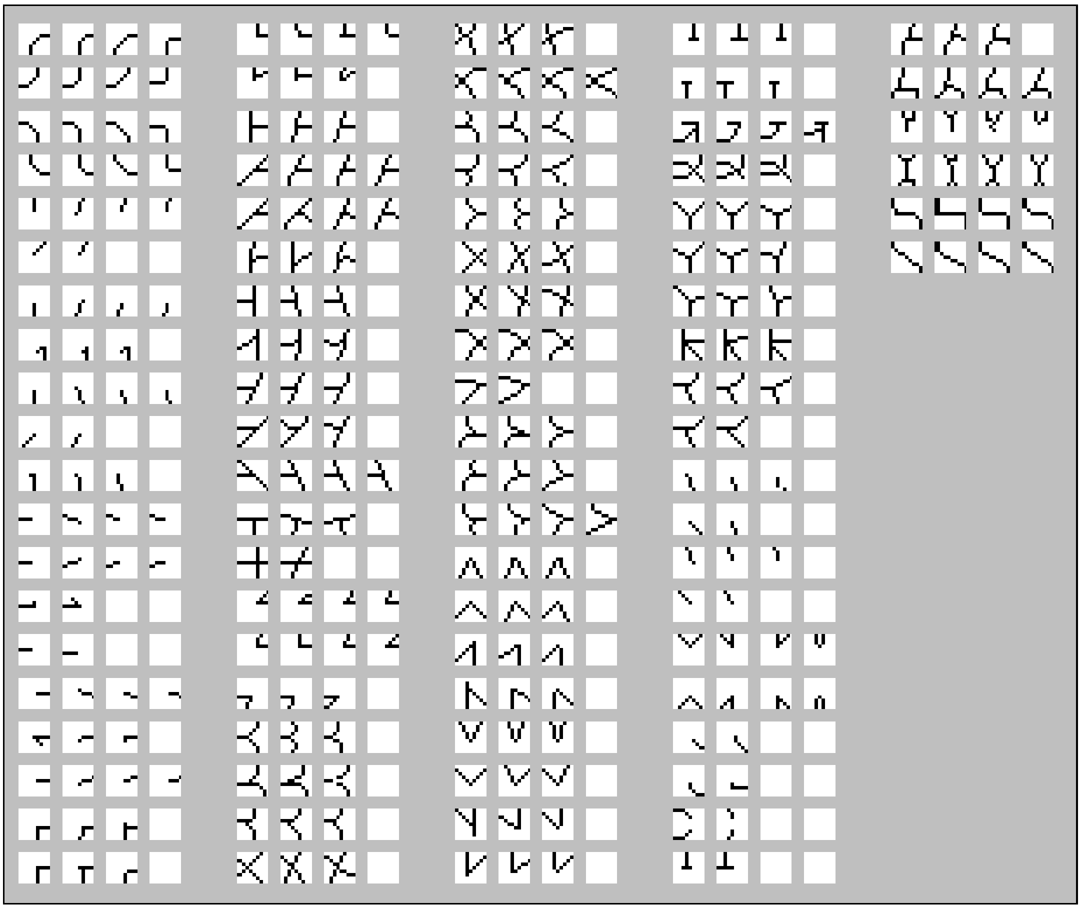

Fig. 2.4.1 Specific training set patterns layer 2 for license plate characters

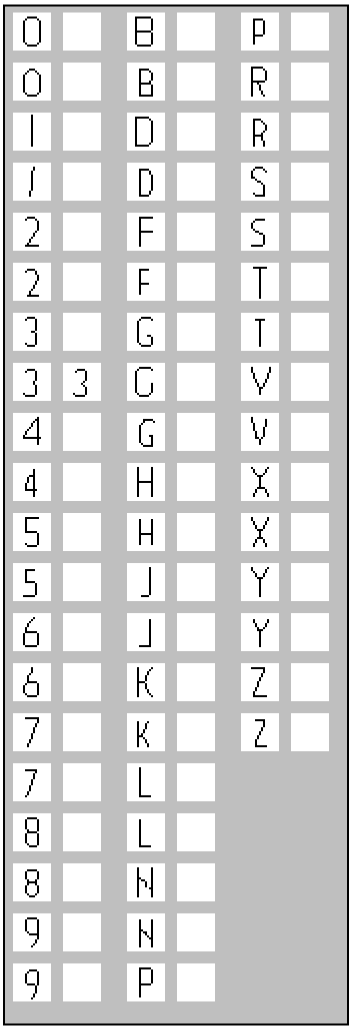

Fig. 2.4.2 Specific training set patterns layer 3 for license plate characters

Table 2.4.1 The 3-layer network configuration for license plate characters

Layer	Plane size	Connectable Area	Number of Planes	Neuron Gap	Training pattern size
Input	19*19				
S-layer 1	19*19	3*3	12	1	3*3
C-layer 1	21*21	3*3	8	1	-
	Selectivity	**Gamma**	**Delta**	**DeltaBar**	
	1.70	0.90	0.90	4.00	
Sublayer inter connections	0 1 2 3 4 5 6 7 8 9 10 11 0 1 1 2 3 3 4 5 5 6 7 7				
S-layer 2	21*21	5*5	86	1	9*9
C-layer 2	13*13	7*7	39	2	-
	Selectivity	**Gamma**	**Delta**	**DeltaBar**	
	3.20	0.90	0.80	4.00	
Sublayer inter connections	00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 00 01 02 03 04 04 05 05 05 05 05 06 06 06 06 07 07 07 08 08 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 09 09 10 10 10 10 11 11 11 11 11 12 13 14 14 15 16 16 16 16 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 16 16 16 16 17 17 17 17 17 17 17 17 18 18 18 18 19 19 19 19 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 20 21 22 23 24 24 24 25 25 25 26 26 27 27 28 29 30 30 31 32 80 81 82 83 84 85 33 34 35 36 37 38				
S-layer 3	11*11	11*11	55	1	19*19
Output	1*1	5*5	27	2	-
	Selectivity	**Gamma**	**Delta**	**DeltaBar**	
	1.40	0.90	0.70	1.4	
Sublayer inter connections	0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 0 0 1 1 2 2 3 3 4 4 5 5 6 6 7 7 8 8 9 9 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 10 10 11 11 12 12 13 13 13 14 14 15 15 16 16 17 17 18 18 19 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 19 20 20 21 21 22 22 23 23 24 24 25 25 26 26				

2.5 Neocognitron discriminator networks

2.5.1 0/D discriminator

Editable training set files:

Layer 2: NeoSmallODlayer2	TextDocument	4KB
Layer 3: NeoSmallODlayer3	TextDocument	5 KB

Coded Neocognitron training set file:

meAlphaSmall0D1	TextDocument	8 KB

Runtime Neocognitron network file:

meAlphaSmall0D1	Net file	65 KB

Table 2.5.1 The layer network configuration for 0/D discriminator

Layer	Plane size	Connectable Area	Number of Planes	Neuron Gap	Training pattern size
Input	19*19				
S-layer 1	19*19	3*3	12	1	3*3
C-layer 1	21*21	3*3	12	1	-
	Selectivity	**Gamma**	**Delta**	**DeltaBar**	
	1.70	0.90	0.90	4.00	
Sublayer inter connections	0 1 2 3 4 5 6 7 8 9 10 11 0 1 2 3 4 5 6 7 8 9 10 11				
S-layer 2	21*21	5*5	8	1	9*9
C-layer 2	13*13	7*7	8	2	-
	Selectivity	**Gamma**	**Delta**	**DeltaBar**	
	3.20	0.90	0.80	4.00	
Sublayer inter connections	00 01 02 03 04 05 06 07 00 01 02 03 04 05 06 07				
S-layer 3	11*11	11*11	4	1	19*19
Output	1*1	5*5	2	2	-
	Selectivity	**Gamma**	**Delta**	**DeltaBar**	
	1.30	0.90	0.70	1.4	
Sublayer inter connections	00 01 02 03 00 00 01 01				

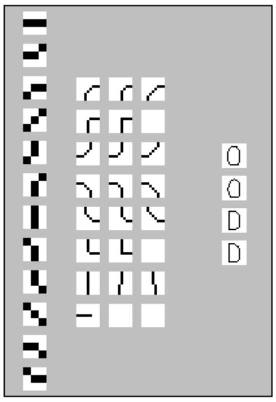

Fig. 2.5.1 0/D discriminator network training set

2.5.2 1/J discriminator

Editable training set files:

Layer 2: NeoSmalllJlayer2	TextDocument	2 KB
Layer 3: NeoSmall1Jlayer3	TextDocument	5 KB

Coded Neocognitron training set file:

MeAlphaSmall1J1	TextDocument	6 KB

Runtime Neocognitron network file:

MeAlphaSmall1J1	Net file	41 KB

Table 2.5.2 The layer network configuration for 1/J discriminator

Layer	Plane size	Connectable Area	Number of Planes	Neuron Gap	Training pattern size
Input	19*19				
S-layer 1	19*19	3*3	12	1	3*3
C-layer 1	21*21	3*3	12	1	-
	Selectivity	**Gamma**	**Delta**	**DeltaBar**	
	1.70	0.90	0.90	4.00	
Sublayer inter connections	0 1 2 3 4 5 6 7 8 9 10 11 0 1 2 3 4 5 6 7 8 9 10 11				
S-layer 2	21*21	5*5	4	1	9*9
C-layer 2	13*13	7*7	4	2	-
	Selectivity	**Gamma**	**Delta**	**DeltaBar**	
	3.20	0.90	0.80	4.00	
Sublayer inter connections	00 01 02 03 04 00 01 02 03 04				
S-layer 3	11*11	11*11	4	1	19*19
Output	1*1	5*5	2	2	-
	Selectivity	**Gamma**	**Delta**	**DeltaBar**	
	1.30	0.90	0.70	1.4	
Sublayer inter connections	00 01 02 03 00 00 01 01				

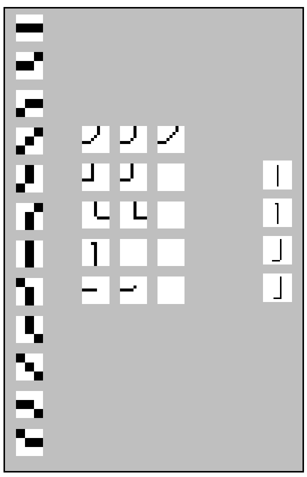

Fig. 2.5.2 1/J discriminator network training set

2.5.3 2/Z discriminator

Editable training set files:

Layer 2: NeoSmall2Zlayer2	TextDocument	3 KB
Layer 3: NeoSmall2Zlayer3	TextDocument	5 KB

Coded Neocognitron training set file:

MeAlphaSmall2Z1	TextDocument	7 KB

Runtime Neocognitron network file:

MeAlphaSmall2Z1	Net file	50 KB

Table 2.5.3 The layer network configuration for 2/Z discriminator

Layer	**Plane size**	**Connectable Area**	**Number of Planes**	**Neuron Gap**	**Training pattern size**
Input	19*19				
S-layer 1	19*19	3*3	12	1	3*3
C-layer 1	21*21	3*3	12	1	-
	Selectivity	**Gamma**	**Delta**	**DeltaBar**	
	1.70	0.90	0.90	4.00	
Sublayer inter connections	0 1 2 3 4 5 6 7 8 9 10 11 0 1 2 3 4 5 6 7 8 9 10 11				
S-layer 2	21*21	5*5	6	1	9*9
C-layer 2	13*13	7*7	6	2	-
	Selectivity	**Gamma**	**Delta**	**DeltaBar**	
	3.20	0.90	0.80	4.00	
Sublayer inter connections	00 01 02 03 04 05 00 01 02 03 04 05				
S-layer 3	11*11	11*11	4	1	19*19
Output	1*1	5*5	2	2	-
	Selectivity	**Gamma**	**Delta**	**DeltaBar**	
	1.30	0.90	0.70	1.4	
Sublayer inter connections	00 01 02 03 00 00 01 01				

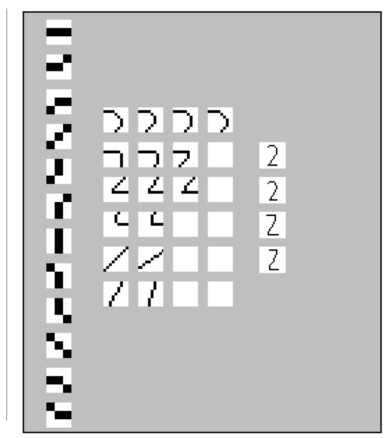

Fig. 2.5.3 2/Z discriminator network training set

2.5.4 5/S discriminator

Editable training set files:

Layer 2: NeoSmall5Slayer2	TextDocument	5 KB
Layer 3: NeoSmall5Slayer3	TextDocument	5 KB

Coded Neocognitron trainingset file:

MeAlphaSmall5S1	TextDocument	9 KB

Runtime Neocognitron network file:

MeAlphaSmal5S1	Net file	88 KB

Table 2.5.4 The layer network configuration for 5S discriminator

Layer	Plane size	Connectable Area	Number of Planes	Neuron Gap	Training pattern size
Input	19*19				
S-layer 1	19*19	3*3	12	1	3*3
C-layer 1	21*21	3*3	12	1	-
	Selectivity	**Gamma**	**Delta**	**DeltaBar**	
	1.70	0.90	0.90	4.00	
Sublayer inter connections	0 1 2 3 4 5 6 7 8 9 10 11 0 1 2 3 4 5 6 7 8 9 10 11				
S-layer 2	21*21	5*5	10	1	9*9
C-layer 2	13*13	7*7	10	2	-
	Selectivity	**Gamma**	**Delta**	**DeltaBar**	
	3.20	0.90	0.80	4.00	
Sublayer inter connections	00 01 02 03 04 05 06 07 08 09 00 01 02 03 04 05 06 07 08 09				
S-layer 3	11*11	11*11	4	1	19*19
Output	1*1	5*5	2	2	-
	Selectivity	**Gamma**	**Delta**	**DeltaBar**	
	1.30	0.90	0.70	1.4	
Sublayer inter connections	00 01 02 03 00 00 01 01				

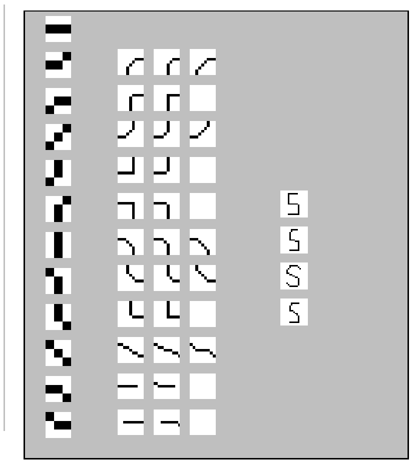

Fig. 2.5.4 5/S discriminator network training set

2.5.5 8/B discriminator

Editable training set files:

Layer 2: NeoSmall8Blayer2	TextDocument	4 KB
Layer 3: NeoSmall8Blayer3	TextDocument	5 KB

Coded Neocognitron training set file:

MeAlphaSmall8B1	TextDocument	7 KB

Runtime Neocognitron network file:

MeAlphaSma8B1	Net file	73 KB

Table 2.5.5 The layer network configuration for 8/B discriminator

Layer	Plane size	Connectable Area	Number of Planes	Neuron Gap	Training pattern size
Input	19*19				
S-layer 1	19*19	3*3	12	1	3*3
C-layer 1	21*21	3*3	12	1	-
	Selectivity	**Gamma**	**Delta**	**DeltaBar**	
	1.70	0.90	0.90	4.00	
Sublayer inter connections	0 1 2 3 4 5 6 7 8 9 10 11 0 1 2 3 4 5 6 7 8 9 10 11				
S-layer 2	21*21	5*5	9	1	9*9
C-layer 2	13*13	7*7	9	2	-
	Selectivity	**Gamma**	**Delta**	**DeltaBar**	
	3.20	0.90	0.80	4.00	
Sublayer inter connections	00 01 02 03 04 05 06 07 08 00 01 02 03 04 05 06 07 08				
S-layer 3	11*11	11*11	4	1	19*19
Output	1*1	5*5	2	2	-
	Selectivity	**Gamma**	**Delta**	**DeltaBar**	
	1.30	0.90	0.70	1.4	
Sublayer inter connections	00 01 02 03 00 00 01 01				

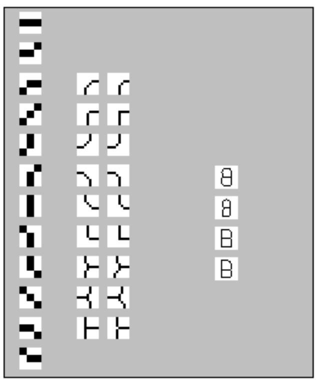

Fig. 2.5.5 8/B discriminator network training sets

2.6 Three layered network used for system evaluation

This network is decribed in section 4.4 of the main document.

Editable training set files:

Layer 2: NeoPatsLayer2Cornet17	TextDocument	33KB
Layer 3: NeoPatsLayer3Cornet17	TextDocument	70 KB

Coded Neocognitron training set file:

meAlphaMaxNewPatternsLayer17	TextDocument	94 KB

Runtime Neocognitron network file:

FukusimaMaxPatternsAlphaLayer17	Net file	2,752 KB

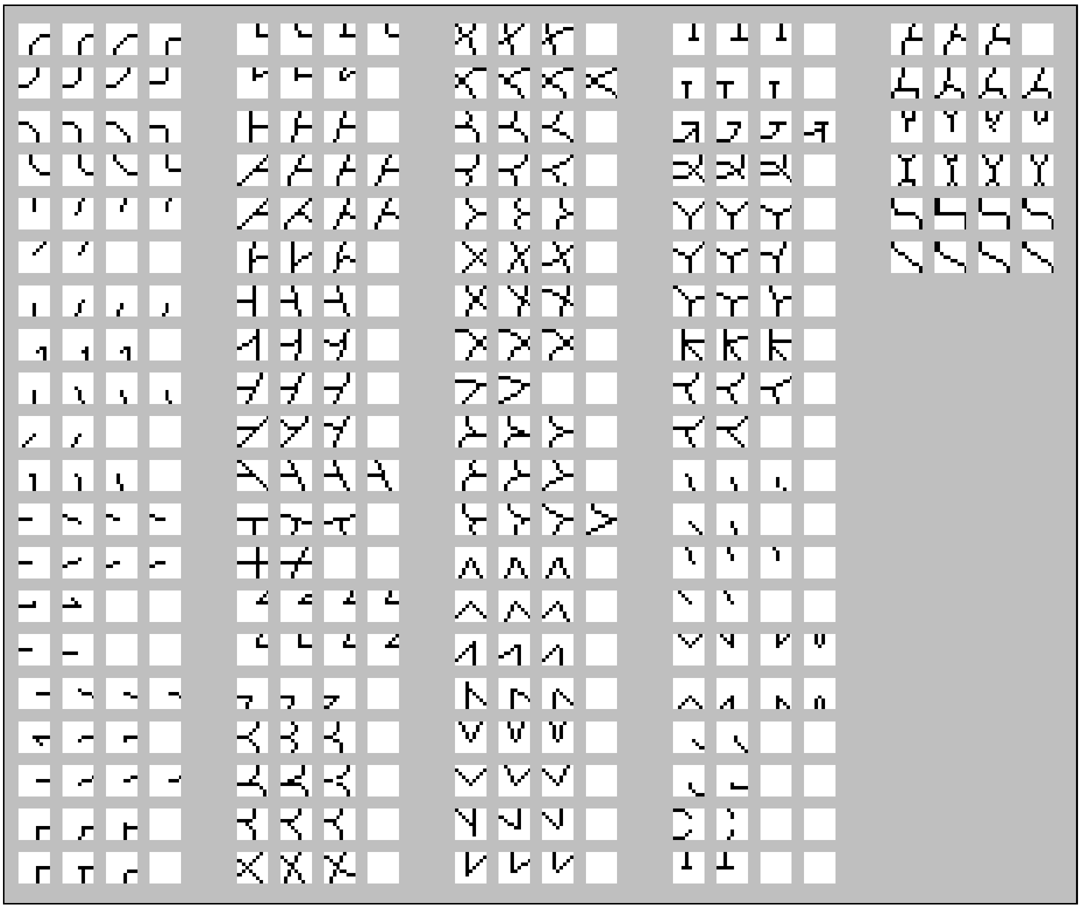

Fig. 2.6.1 Trainingset patterns layer 2 for license plate characters new style

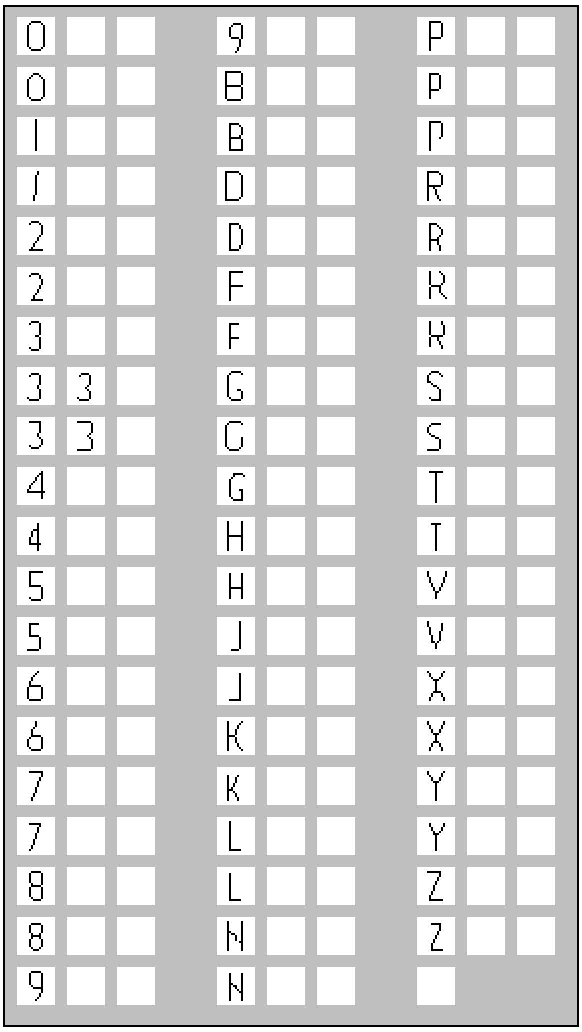

Fig. 2.6.2 Trainingset patterns layer 3 for license plate characters new style

Table 2.6.1 The 3-layer network configuration for license plate characters

Layer	Plane size	Connectable Area	Number of Planes	Neuron Gap	Training pattern size
Input	19*19				
S-layer 1	19*19	3*3	12	1	3*3
C-layer 1	21*21	3*3	8	1	-
	Selectivity	**Gamma**	**Delta**	**DeltaBar**	
	1.70	0.90	0.90	4.00	
Sublayer inter connections	0 1 2 3 4 5 6 7 8 9 10 11 0 1 1 2 3 3 4 5 5 6 7 7				
S-layer 2	21*21	5*5	86	1	9*9
C-layer 2	13*13	7*7	39	2	-
	Selectivity	**Gamma**	**Delta**	**DeltaBar**	
	3.20	0.90	0.80	4.00	
Sublayer inter connections	00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 00 01 02 03 04 04 05 05 05 05 05 06 06 06 06 07 07 07 08 08 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 09 09 10 10 10 10 11 11 11 11 11 12 13 14 14 15 16 16 16 16 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 16 16 16 16 17 17 17 17 17 17 17 17 18 18 18 18 19 19 19 19 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 20 21 22 23 24 24 24 25 25 25 26 26 27 27 28 29 30 30 31 32 80 81 82 83 84 85 33 34 35 36 37 38				
S-layer 3	11*11	11*1	59	1	19*19
Output	1*1	5*5	27	2	-
	Selectivity	**Gamma**	**Delta**	**DeltaBar**	
	1.40	0.90	0.70	1.4	
Sublayer inter connections	00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 00 00 01 01 02 02 03 03 03 04 04 05 05 06 06 07 07 08 08 09 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 09 10 10 11 11 12 12 13 13 13 14 14 15 15 16 16 17 17 18 18 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 19 19 19 20 20 20 20 21 21 22 22 23 23 24 24 25 25 26 26				

Appendix III

Results on character recognition

3.1 Fukisuma’s network configuration for handwritten character recognition

The tables in this section contains the recognition results on the network defined in section 2.1 of this appendix. The input samples are binary images of printed characters with various font styles en sizes.

Table 3.1.1

Font Size Character	Arial 20pt	18pt	16pt	14pt	12pt	10pt	8pt	Recog. Correct
0	Q9S00	Q90SC	Q9C0S	Q90SC	0SCQ9	DPR00	DPR50	14
1	1M400	1M400	7T000	10000	M0000	M0000	M0000	43
2	2SQ00	2SD00	2S0Q0	20000	2Z000	2Z000	2Z000	100
3	Q9200	30000	-	-	B0000	-	-	14
4	00000	00000	40000	40000	40000	40000	00000	57
5	50000	50000	50000	-	56000	5FE00	FE000	71
6	S6000	6S0C0	S60Q9	SQ069	60S00	-	50000	29
7	70000	70000	70000	7T000	7T000	70000	7T000	100
8	-	8Q000	C0G36	Q9S00	80000	P0000	50000	29
9	Q90SC	9Q0SC	9QC0S	9QC0S	9Q0SC	B0000	PF000	57
A	A0000	A0000	A0000	A0000	A0000	A0000	A0000	100
B	SI000	DIS00	B0000	B0000	DE000	DE500	5D000	29
C	C0Q90	QC09S	C09SQ	CG060	C0Q9G	C0Q9S	CG000	86
D	SD600	SD060	SD060	DS000	DPR50	DPR50	DPR00	57
E	EFL00	EFL00	EFL00	ELF00	EF000	EF000	E0000	100
F	FE000	FE000	FE000	FE000	FE000	FE000	F0000	100
G	GC000	QGC09	GC0Q9	QGC09	QCG09	Q9C0S	Q9C0S	29
H	H4100	H4100	H4100	H4100	H4100	4H100	H0000	86
I	1JY4U	1U400	14000	14000	10000	10000	10000	100
J	1J000	J1U00	JU000	1J000	J0000	J0000	NV000	57
K	K0000	KH140	1K4H0	K1400	K0000	K0000	K0000	86
L	L1E00	L1E00	L1E40	L1E00	L1000	L1400	L0000	100
M	M0000	PF000	PF000	00000	M0000	-	WVN00	29
N	14H00	41000	NVW00	NVW00	NVW00	NV000	N0000	71
P	PSRD6	PSRD6	PRSD0	PSRD0	PRD00	PRD00	P0000	100
R	R0000	RSP00	S9RQ0	RP000	RP000	RP000	RK000	86
S	CS0Q9	S0000	C0000	-	S0C69	DPR50	P0000	29
T	T0000	T0000	T0000	T0000	T0000	T0000	T0000	100
U	U0000	U0000	UJ000	U1000	U0000	U0000	U1000	100
V	VWN00	U1000	41U00	V1UN0	VNW00	VNW00	VNW00	71
W	WVN00	U4H10	40000	WVN00	WVN00	WVN00	WNV00	71
X	-	14000	-	K0000	XK000	X0000	X0000	43
Y	Y0000	Y0000	Y0000	Y1000	Y1000	VN000	Y0000	86
Z	LE000	LEZ00	ZL2E0	ZL000	Z0000	Z0000	Z0000	71
Recognised Correctly	62	65	62	71	88	65	62	**68**

Table 3.1.2

Font Size Character	Arial Bold 20pt	18pt	16pt	14pt	12pt	10pt	8pt	Recognised Correctly
0	DPSR0	0C9SQ	Q09CS	-	0CQ9S	-	-	71
1	T0000	H4000	H4100	70000	70000	70000	-	0
2	2DS00	20000	LE000	2L000	L0000	L0000	Z0000	43
3	-	-	B0000	-	-	-	-	0
4	-	40000	-	41000	-	14000	-	29
5	5FE00	50000	50000	50000	50000	E0000	FE500	71
6	65000	6S000	60000	S6000	-	FE000	50000	43
7	7T000	7T000	7ZT00	70000	70000	70000	7T000	100
8	P5RD0	Q90SC	30000	Q9000	-	-	-	0
9	-	9QC00	Q0000	00000	-	50000	-	14
A	-	A0000	A0000	A0000	A0000	-	-	57
B	D0000	DPRB0	DPE00	DE000	DI000	-	-	0
C	F0000	CG000	CG000	CS000	C9000	C0000	C0000	86
D	DS000	DPR00	DPR00	DPR50	DSI00	DPR50	-	86
E	EFL00	EFL00	EFL00	EF000	EL000	EF000	EF000	100
F	FE000	FE000	FE000	FE000	FE000	FE000	F0000	100
G	C0000	GC000	GC000	QGC09	Q9C0G	C0000	-	29
H	H4100	H4100	H4100	H4100	4H100	H0000	-	71
I	1U000	1U000	1J000	10000	14000	14000	14000	100
J	41J00	J1000	J1400	J0000	J0000	40000	4H100	57
K	K1000	H4100	41H00	K4000	4K000	K0000	41H00	43
L	L14E0	L1E00	L1E40	L1E00	L1E00	L0000	L1000	100
M	YH100	H0000	M0000	M0000	M0000	1H400	4H100	43
N	1N400	14YH0	14N00	N1000	NV000	10000	40000	29
P	P0000	PRD00	PRD00	PRD00	PRS00	-	-	71
R	A0000	RPDF0	RPF00	RPF00	RPK00	-	-	57
S	50000	S60D5	SD600	S6D00	S0000	50000	-	57
T	T0000	T0000	T0000	T0000	T0000	T0000	T0000	100
U	U1000	U1400	U1000	U1000	U0000	U1000	1U400	86
V	U1400	10000	VU1N4	VN000	VN000	VN000	NV000	57
W	NW000	Y0000	4H100	WVN00	WVN00	WVN00	41000	43
X	41HM0	00000	XK000	K0000	K0000	10000	50000	14
Y	Y1000	Y1000	Y0000	Y0000	Y1000	P0000	Y0000	86
Z	ZL2E0	ZLE00	Z0000	Z0000	L0000	Z0000	Z0000	86
Recognised Correctly	47	71	68	76	62	44	29	**57**

Table 3.1.3

Font Size Character	Veranda 20pt	18pt	16pt	14pt	12pt	10pt	8pt	Recognised Correctly
0	Q0SC9	Q9SC0	0QCS9	Q90SC	0SCQ9	DPR50	DPR50	29
1	-	-	-	-	I1000	-	-	14
2	2LES0	2LE00	2LSDE	20000	2S000	20000	Z0000	86
3	-	B0000	B0000	B0000	3B000	-	-	14
4	-	41I00	-	-	40000	-	40000	43
5	50000	50000	50000	50000	50000	F5000	FE000	71
6	605S0	605S0	65000	50000	65000	-	-	57
7	7T000	70000	7T000	70000	70000	70000	70000	100
8	90000	8Q3S9	80000	389S0	83000	00000	50000	43
9	Q9S00	Q9C0S	Q9C0S	9QCS0	Q90SC	PRD00	-	14
A	AM000	A0000	A0000	A0000	AM000	40000	A0000	86
B	DP000	B0000	B0000	00000	DPBR0	D0000	5D000	29
C	C0000	C2000	CQ0G9	CG000	CG000	CG000	CG000	100
D	DPISR	SD060	SD0I6	SD060	DPR50	DPR50	DPR50	57
E	EFL00	EFL00	ELF00	EL000	EF000	EFL00	E0000	100
F	FE000	FE000	FE000	FE000	FE000	FE000	F0000	100
G	00000	G0000	GC000	GC000	GC000	CG000	CG000	57
H	H4100	H4100	H4100	H4100	H4100	4H100	H0000	86
I	LE000	I1T00	1LIU4	14L00	I0000	10000	10000	86
J	-	-	-	-	-	-	-	0
K	K1400	K0000	K1400	K14H0	K0000	K0000	K0000	100
L	L1E00	L1E00	L1E40	L1E00	L1E00	L1400	L0000	100
M	MVN00	PF000	P0000	MP000	M0000	M0000	-	57
N	Y1400	VNW00	41P00	P0000	NVW00	N0000	N0000	43
P	PDRS0	PSDR0	PRSD0	PSRD9	PRD00	PRD00	P0000	100
R	RPD00	R0000	RK000	RKP00	RPK00	RK000	RP000	100
S	S0000	C0000	50000	50000	S0CQ9	5F000	P0000	29
T	T0000	T0000	T0000	T0000	T0000	T0000	T0000	100
U	U1000	U0000	U1000	U1000	U0000	U0000	U1000	100
V	41U00	VWNU0	VWN00	VN000	VNW00	VNW00	VNW00	86
W	WVN00	NVW00	-	U0000	VWN00	NVW00	WVN00	29
X	JK000	KX100	K1400	KV100	XK000	X0000	X0000	43
Y	Y0000	Y1000	Y1000	Y1000	Y1000	VN000	Y0000	86
Z	LE200	LEZ00	LE000	ZLE00	Z2000	Z0000	Z0000	57
Recognised Correctly	59	65	65	62	88	53	62	**65**

Table 3.1.4

Font Size Character	Veranda Bold 20pt	18pt	16pt	14pt	12pt	10pt	8pt	Recognised Correctly
0	0CSQ6	CQ09S	0QS9C	0SCQ9	0CS96	-	-	86
1	-	-	-	Z0000	-	-	14000	14
2	20000	20000	2Z000	20000	2Z000	Z0000	Z2000	71
3	-	-	-	B0000	B0000	-	-	0
4	-	I1000	-	-	40000	40000	40000	43
5	50000	50000	5FE00	5FE00	5F000	5EF00	FE500	86
6	50000	60000	5FE00	50000	50000	EF000	50000	14
7	7T000	T7000	70000	-	7T000	70000	7T000	71
8	-	80000	-	RPB00	00000	-	50000	14
9	-	-	Q90SC	DPR00	9Q000	-	-	14
A	PR000	70000	A0000	P0000	A0000	A0000	-	43
B	-	PRF00	PRB00	PDR00	P0000	-	-	0
C	CG000	QC90S	CQ09S	C0000	C0000	C0000	-	71
D	SD600	DPRS0	DPSR0	DPSR0	DSPR0	DPR50	DPR00	86
E	EL000	EF000	EF000	EF000	EF000	EF000	EF000	100
F	FE000	FE000	FE000	FE000	FE000	FE000	F0000	100
G	GC000	CGQ09	GC000	CG000	GC000	C0000	-	43
H	H4100	H4100	H4100	4H100	H4100	4H000	-	57
I	I1000	00000	T0000	00000	00000	14H00	14000	43
J	-	-	-	-	-	-	-	0
K	K0000	14H00	K14H0	14H00	41000	1H000	50000	29
L	L1E40	L1400	L1000	L1000	L1000	L0000	L1000	100
M	MH100	H14M0	MH100	M0000	M0000	M0000	-	71
N	Y1M00	K4000	41000	4N1H0	N1V40	N1000	N0000	43
P	PRD00	PRD00	PR000	PRD00	PRD00	00000	-	71
R	PR000	PDR00	RPK00	RP000	P0000	-	-	29
S	5F000	S0000	S0000	50000	50000	50000	P0000	29
T	T0000	TF000	T0000	T0000	T0000	T0000	T0000	100
U	U0000	U1400	U0000	U1000	U0000	U1400	1U400	86
V	10000	U1000	U1400	41U00	14UJ0	14000	VWN00	14
W	00000	W0000	4U100	NWV00	NVW00	VNW00	41000	14
X	41000	10000	K1400	KX000	-	X0000	50000	14
Y	Y0000	Y1000	Y0000	Y0000	Y1000	PFR00	PFR00	71
Z	LEZ00	LZE00	LEZ00	LZ000	ZL000	Z0000	Z0000	43
Recognised Correctly	50	44	59	41	62	50	38	**49**

Table 3.1.5

Font Size Character	EuroStile 20pt	18pt	16pt	14pt	12pt	10pt	8pt	Recognised Correctly
0	QG0S9	Q09CG	0CQS9	06SC9	DRPS0	DPR50	50000	29
1	70000	70000	70000	70000	-	M0000	M0000	0
2	L2E00	2SQL0	2LE00	LE200	DER00	EF000	Z0000	29
3	-	-	-	B0000	PRD00	DPF00	-	0
4	I0000	-	-	-	-	40000	4M000	29
5	5F000	50000	5FE00	5D000	5FE00	5EF00	-	86
6	6S000	65000	-	36000	PRD00	P0000	50000	29
7	70000	70000	7T000	7T000	7T000	70000	70000	100
8	80000	80000	80000	PRD00	DPR00	-	-	43
9	Q90CS	Q90CS	9QC00	-	DPR00	50000	PF000	14
A	PRD00	-	A0000	A0000	A0000	A0000	A0000	71
B	I0000	B0000	-	DPR00	DPE00	D0000	-	14
C	40000	QC90S	C0000	C0G00	DPR00	-	EF000	29
D	SID00	IS1D0	DPR00	DPR00	DPR00	DPR00	DP5R0	71
E	EFL00	EFL00	EF000	EFL00	EF000	EFL00	E5F00	100
F	FE000	F0000	FE000	FEP00	FE000	F0000	P0000	86
G	G0000	G0000	G0000	G0000	00000	DPR50	EF500	57
H	H4100	H4100	H4100	H4100	H4100	4H100	4H000	71
I	1U400	14000	14000	1J000	10000	10000	10000	43
J	U1J00	J1000	J1000	J1U00	J1000	J0000	41000	71
K	HU100	K0000	K0000	14H00	10000	-	K0000	43
L	L1E00	L1E40	L1000	L1E40	L1000	L1000	1L000	86
M	PFR00	PFMR0	PF000	00000	W0000	WVN00	-	0
N	VNW00	41000	P1000	1N4V0	NVW00	N0000	N0000	43
P	PD600	00000	P0000	PRD00	PRD00	PRD00	P0000	86
R	-	9SQ00	7Y000	PRF00	RPD00	PR000	-	14
S	S0000	Q9SC0	-	-	D0000	50000	PDF00	14
T	T0000	T0000	T0000	T0000	T0000	T0000	T0000	100
U	U1000	U1000	U1000	U1400	U1400	U1400	41000	86
V	U0000	VWN00	14000	VN000	VWN00	VWN00	VWN00	71
W	10000	NV000	00000	H0000	00000	WVN00	WVN00	29
X	X0000	X0000	X0000	K0000	XK000	00000	X0000	71
Y	Y0000	Y1000	Y1000	Y0000	VN000	Y0000	Y0000	86
Z	LEZ00	Z2L00	ZLE00	Z0000	Z0000	Z0000	Z0000	86
Recognised Correctly	47	59	65	56	50	50	41	**53**

Table 3.1.6

Font Size Character	EuroStile Bold 20pt	18pt	16pt	14pt	12pt	10pt	8pt	Recognised Correctly
0	Q09SC	0QCS9	QS092	CG090	QC900	-	DPRS0	29
1	70000	70000	7T000	70000	70000	70000	70000	0
2	DSEL2	2EL00	2QS0L	LE000	20000	Z0000	Z0000	43
3	380S0	Q9200	Q9S0C	-	2Q000	-	-	14
4	I0000	D0000	-	-	-	-	-	0
5	5EF00	50000	50000	5E000	E5000	5FE00	-	71
6	65000	S6Q00	6D000	90000	00000	EF500	E5000	29
7	7T000	70000	7T000	7T000	70000	70000	7T000	100
8	DPR00	Q8900	-	Q8900	Q9000	50000	-	0
9	5D000	Q9S02	Q9000	Q9C00	Q9000	-	-	0
A	DP000	A0000	PRF00	-	-	-	-	14
B	PDRE0	DPR00	-	I0000	-	-	50000	0
C	CG000	CG000	CQ900	CQ90G	20000	FE000	50000	57
D	DPRS0	DPSR0	DPI00	DSI00	D0000	00000	DP5R0	86
E	EFL00	EFL00	EFL00	EL000	EL000	EF000	EF500	100
F	FE000	FE000	FE000	FE000	F0000	FE000	F0000	100
G	GC000	C0000	CG000	QGC09	Q9C0G	F0000	EF500	14
H	H4100	H4100	H4100	H4100	4H100	4H000	4H000	43
I	1U000	1J000	10000	14000	14H00	14000	10000	86
J	UJ100	J1000	41J00	J1000	41000	40000	14000	29
K	1H400	K0000	14K00	K1400	K0000	K0000	14000	57
L	L1E00	L1E00	L1000	L1000	L1400	L0000	1L000	86
M	M4000	-	WNV00	P0000	00000	M4000	P0000	29
N	VNW00	14000	14H00	VWN00	VNW00	-	-	0
P	PRDS0	PRD00	PRDS0	P0000	P9R00	-	P0000	86
R	RPDF0	RP000	RPD00	RPD00	S0000	-	-	57
S	56F00	SD000	-	-	-	E5000	FE500	14
T	T0000	T0000	T0000	T0000	T0000	T0000	T0000	100
U	U0000	U14J0	U1400	U1000	41U00	U1000	U1400	86
V	U1400	1U4J0	14U00	VWN00	VWN00	41000	N0000	29
W	WVN00	10000	WVN00	-	WV000	-	41000	43
X	-	K1400	KX000	XK000	X0000	-	WVN00	29
Y	Y1000	Y1000	Y1000	Y1000	Y0000	Y0000	Y0000	100
Z	LEZ00	LEZ00	ZLE00	ZL000	Z0000	-	Z0000	57
Recognised Correctly	56	56	53	56	44	35	26	**47**

Table 3.1.7

Font Size Character	License Plate Font 22px	18px	16px	14px	12px	10px	Recognised Correctly
0	Q9C00	Q9S0C	Q0SC9	Q02S9	SD600	D5000	0
1	1J400	1J400	1J400	70000	14H00	10000	83
2	L2E00	-	20000	-	-	-	17
3	-	-	-	-	-	-	0
4	-	41000	4H000	40000	40000	4H100	83
5	5EF00	50000	5E000	50000	50000	-	83
6	41000	60SC0	60SC0	41000	10000	14000	33
7	7T000	70000	70000	70000	7T000	70000	100
8	-	PB000	Q9000	Q0000	-	-	0
9	9QC00	Q9C00	Q9C00	9Q000	-	-	33
B	PR000	DPR00	DPR00	D0000	-	-	0
D	DRPS0	DIS00	DPR00	D0000	SD600	DPR50	83
F	FE000	F0000	F0000	-	-	-	50
G	00000	40000	40000	40000	CG000	DE500	0
H	H4100	4H100	4H100	H4100	4H100	1H400	33
J	J0000	00000	J0000	10000	J1000	40000	50
K	14K00	KX000	K0000	K0000	K0000	K0000	83
L	L1E00	L1E00	L14E0	L1U00	L1400	L0000	100
N	N14H0	N14H0	N1400	N0000	N0000	14000	83
P	PRD00	PRD00	P0000	PR000	P0000	P0000	100
R	RPK00	RPK00	-	RP000	R0000	00000	67
S	-	-	-	6S000	50000	P0000	0
T	T7000	M1400	TY000	7Y000	10000	10000	33
V	Y1000	VYN00	VN000	VWN00	VNW00	N0000	67
X	14K00	K1400	V0000	VN000	-	VN000	0
Y	Y1000	Y1000	14HY0	Y0000	Y0000	V0000	67
Z	LE000	LE000	L0000	ZL000	L0000	LE000	17
Recognised Correctly	51.85	51.85	51.85	55.56	44.44	25.93	

Table 3.1.8 Absolute recognition figures of printed characters using the original Fukisuma network designed for handwritten characters

Character input	recognized as 0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	G	H	I	J	K	L	M	N	P	Q	R	S	T	U	V	W	X	Y	Z	?	!	#	Rates in % classified correctly	mis classified	un classified
0	12					1							2	9												17		1								6	30	48	25.0	62.5	12.5
1		9						16										2	1				5						1						1	13	26	48	18.8	54.2	27.1
2			27											2	1							7													7	4	17	48	56.3	35.4	8.3
3			1	3								8		1											1	3										31	14	48	6.3	29.2	64.6
4	3	1			19									1					3																	21	8	48	39.6	16.7	43.8
5						37									2	5																				4	7	48	77.1	14.6	8.3
6	1	2		1	2	9	16			1					3	1									2			5								5	27	48	33.3	56.3	10.4
7								46																					1							1	1	48	95.8	2.1	2.1
8	2			2		4			9	1			1	2											4	8	1									14	25	48	18.8	52.1	29.2
9	1					3				10		1		2											3	15										13	25	48	20.8	52.1	27.1
A					1			1			26			1											4											9	7	42	61.9	16.7	21.4
B	1					3						5		18					2						6			1								12	31	48	10.4	64.6	25.0
C			1		1	1							30	1	1	2										3										2	10	42	71.4	23.8	4.8
D	1													36					1									9								1	11	48	75.0	22.9	2.1
E															42																					6	0	48	87.5	0.0	12.5
F																44									1											3	1	48	91.7	2.1	6.3
G	3				3								10	2	2	1	16									9										2	30	48	33.3	62.5	4.2
H		1			13													32																		2	14	48	66.7	29.2	4.2
I	3	34																	3			1							1							0	39	42	7.1	92.9	0.0
J	1	4			8															18				1						2						14	16	48	37.5	33.3	29.2
K		10			4	1												2			30															1	17	48	62.5	35.4	2.1
L		2																				46														0	2	48	95.8	4.2	0.0
M	3	1			1													2					16		9							4		1		5	21	42	38.1	50.0	11.9
N		9			6																1			21	2						5			2		2	25	48	43.8	52.1	4.2
P	2																								42											4	2	48	87.5	4.2	8.3
R	1							1		1	1														5		28	2								9	11	48	58.3	22.9	18.8
S						12	1						3	2	1	1									5	1		12								10	26	48	25.0	54.2	20.8
T		2						1															1						44							0	4	48	91.7	8.3	0.0
U		2			2																									38						0	4	42	90.5	9.5	0.0
V		7			4																			3						6	27			1		0	21	48	56.3	43.8	0.0
W	3	2			6													1						6						2	2	16		1		3	23	42	38.1	54.8	7.1
X	2	4			2	2														1	12										3	1	15			6	27	48	31.3	56.3	12.5
Y		1																							3						4			40		0	8	48	83.3	16.7	0.0
Z																						18													29	1	18	48	60.4	37.5	2.1

?: number of characters unclassified
!: number of characters misclassified
#: total number of characters

3.2 Fukisuma’s network configuration for handwritten character recognition using a refined training set

The tables in this section contains the recognition results on the network defined in section 2.2 of this appendix. The input samples are binary images of printed characters with various font styles en size.

Table 3.2.1

Font Size Character	Arial 20pt	18pt	16pt	14pt	12pt	10pt	8pt	Recognised Correctly
0	Q98S0	Q980S	Q98C0	Q90S8	0SCQ6	DPRB0	DP000	14
1	1M400	1M400	10000	H0000	M0000	M0000	M0000	43
2	2ESL0	2QSEL	2SDQ0	20000	2Z000	2ZE00	2Z000	100
3	-	30000	30000	3J000	30000	-	-	57
4	40000	40000	40000	40000	40000	40000	40000	100
5	5SF00	5S000	5SE00	EFS50	56000	5SFE0	FSE50	71
6	QS600	6S0Q0	SQ609	QS0D9	60S00	BP000	B0000	29
7	73000	73000	73000	73000	7T000	70000	70000	100
8	00000	8Q000	C0G00	Q98S0	80300	BP000	00000	29
9	Q980S	Q980S	9QC00	Q9C00	Q90CS	B0000	BP000	14
A	A0000	A0000	AX000	A0000	A0000	A0000	-	86
B	BQDP0	BDQP0	BDP00	BPDR0	BP000	DB000	D0000	71
C	2CE00	QC0S9	C0S00	CG000	C0Q9S	C0Q90	C0000	71
D	DSQ6B	DSQ06	DS06Q	DSQ06	DPSB0	DPS00	DPR00	100
E	EFL00	EFL00	EFL00	EFL00	EFL50	EFL00	E0000	100
F	FES50	FES50	FES00	FES00	FE000	FE000	FE000	100
G	00000	QG0C9	C0Q9G	QG0C9	GQC00	Q90CS	Q9C00	14
H	H1000	H1000	H1000	H1000	H1000	H1000	H1000	100
J	1JU00	J1U00	J1U00	J1U00	1UJ00	1J000	NK000	43
K	X0000	XK100	X1K00	XK100	KX100	KX000	KX100	43
L	L1E2U	L1E2U	L1E20	L1E20	L1E20	L1E20	LZ000	100
M	M0000	BP000	PMB00	00000	M0000	M0000	W0000	43
N	10000	10000	1N000	1N000	N0000	N0000	N0000	43
P	PRDSB	PRDSB	PRDSQ	PRDS0	PRDB0	PRB00	P0000	100
R	RPQB0	RPXQB	XRQB0	RPX00	RPB00	RP000	R0000	86
S	SC0Q9	SQ000	C0000	3JS50	S06C9	DPBR0	BP000	43
T	T0000	T7000	T0000	T0000	T7000	T0000	T0000	100
U	U1J00	U1J00	UJ100	1UJV0	UJ100	U1J00	1UJ00	71
V	VWN00	1UVJ0	1UVJ0	1VUK0	VNK00	VNKW0	KNW00	43
W	NW000	U1V00	N0000	WVK00	WN000	WN000	WN000	57
X	X1K00	XK000	XK000	XK000	XK000	X0000	X0000	100
Y	Y1000	Y1000	Y0000	Y1000	Y1000	K0000	Y0000	86
Z	2ZLE7	2ZLE7	2ZE73	Z2EL0	Z2EL0	Z2LE0	Z2000	57
Recognised Correctly	64	70	64	61	91	67	55	**67**

Table 3.2.2

Font Size Character	Arial Bold 20pt	18pt	16pt	14pt	12pt	10pt	8pt	Recognised Correctly
0	DBQSP	0CQS8	Q09S8	-	0CQS9	BP000	-	57
1	73J00	H0000	H0000	17000	-	-	-	14
2	2LEDS	23EL0	2Z3LE	2LZ00	2LEZ0	Z2LE0	Z0000	71
3	3J000	3J000	30000	-	-	30000	-	57
4	40000	40000	40000	40000	-	40000	-	71
5	5FSE0	5SE00	5S000	S5000	-	5SEF0	EF5S0	57
6	65000	6S000	56000	6SQ0D	-	F0000	-	43
7	73000	73T00	73000	70000	70000	70000	70000	100
8	BPD00	Q980S	-	8Q000	00000	B0000	B0000	14
9	BD000	9Q000	QJ000	-	-	5SJ30	-	14
A	B0000	A0000	A0000	4A000	A0000	40000	B0000	43
B	BD000	BPRD0	BP000	BD000	DB000	B0000	00000	71
C	FE000	F5SG0	FCE00	2FE00	CQ000	FEC00	FEL00	14
D	DS000	DSBP0	DB000	DP000	DSP00	DPR00	00000	86
E	EFL00	EFL00	EFL00	EFL00	EFL20	EF000	EF000	100
F	FES00	FE5S0	FE000	FE000	FES00	FES50	FE000	100
G	F5SE0	CG000	GC000	GC000	Q90C8	CG000	FES50	29
H	H1000	H1000	H1000	H1000	H1000	H1000	00000	86
J	1JU00	1JU00	1JU00	J3100	J1000	J1000	1JH00	43
K	KX1R0	1H000	1H000	K1X00	XK100	1KX00	1J000	29
L	L1E20	L1E2U	L1E20	L1E20	LE21Z	L1E00	L1000	100
M	Y1000	-	M0000	MN000	M0000	H0000	-	43
N	1JN00	1Y000	1N000	N1000	N0000	10000	-	29
P	PR000	PRDBS	PRDB0	PRDB0	PRS00	P0000	BP000	86
R	PR000	RPB00	RP000	PRB00	RP000	B0000	BP000	43
S	S5F00	S60D5	SD60Q	SD600	S0000	S5300	S5000	100
T	T0000	T7000	T7000	T7300	T0000	T0000	T0000	100
U	U1JV0	U1J00	U1J00	U1J00	U1J00	1UJ00	1UJ00	71
V	1UJV0	J1000	1UVJK	VK1N0	VK1N0	KN000	KN000	29
W	NW000	10000	HVU10	WNV00	WV000	WN000	10000	43
X	10000	XK000	XK000	XK100	XK000	10000	-	57
Y	Y1000	Y1000	Y1000	Y1000	Y1000	-	Y0000	86
Z	2ZEL3	2Z7E3	2Z7EL	2ZE7L	Z2LE0	Z2000	Z2000	43
Recognised Correctly	55	67	64	79	70	48	27	**58**

Table 3.2.3

Font Size Character	Veranda 20pt	18pt	16pt	14pt	12pt	10pt	8pt	Recognised Correctly
0	Q0CS9	QCS09	0QCS9	Q90S8	0SQC6	DPR00	DP000	29
1	-	-	LEZ00	ZEL00	H0000	-	-	0
2	2LED0	2LEZ0	2LEDS	2ZE00	2SE00	2EF00	Z2000	86
3	3JS00	3J000	3J000	3J000	30000	00000	-	71
4	40000	40000	40000	-	40000	40000	40000	86
5	5SF30	S53JF	5SE00	5S000	50000	5SFE0	FE5S0	71
6	65000	560S0	5FS60	5SF00	65000	FS500	00000	29
7	73000	73000	73000	73000	73000	70000	70000	100
8	9Q000	8Q300	80000	38000	83000	BP000	-	43
9	Q9JS8	QS93C	Q93C0	Q9000	Q90S8	PBDR0	-	0
A	A4M00	4A000	A0000	A0000	AM400	40000	-	57
B	BPRD0	00000	RP000	R0000	PBDR0	D0000	D0000	14
C	FEC00	2FE00	CEF00	CFE00	FEC00	C0000	C0000	57
D	DSBP0	DSQ06	DSQ06	DSQ06	DPSB0	DPS00	DPR00	100
E	EFL00	EFL00	EFL00	EFL20	EFL00	EFL00	E0000	100
F	FE5S0	FES50	FES00	FES00	FES50	FE000	FE000	100
G	SF500	00000	CG000	CG000	CG000	C0000	C0000	0
H	H1000	H1000	H1000	H1000	H1000	H1000	H1000	100
J	3J7S0	3JS70	J3S50	J3S00	3J000	J3000	J3000	57
K	KX100	XK100	KX100	XK100	KX100	K0000	K0000	71
L	L1E2J	L1E2U	L1E20	L1E20	L1E20	L1E00	LZ000	100
M	M0000	P0000	P0000	M0000	M0000	M0000	-	57
N	1Y000	N1000	1N000	10000	N0000	N0000	N0000	57
P	PDRS0	PRDS0	PRDS0	PDRS0	PRDBS	PRB00	P0000	100
R	RPBX0	XRPBD	RP000	RP000	RP000	R0000	RX000	86
S	S3J00	3JSCQ	S5F00	S5000	S0QC6	5SFE0	BP000	57
T	T7FS0	T7000	T0000	T0000	T7000	T0000	T0000	100
U	U1JV0	U1J00	U1J00	1UJ00	UJ100	U1J00	1UJ00	71
V	1UVJ0	V1U00	VW1UN	V1K00	VKN10	NKVW0	KNV00	57
W	WN000	NWV00	10000	1U000	NWV00	NWK00	WKN00	29
X	KX000	XK000	XK100	KX000	XK000	X0000	X0000	71
Y	Y1000	Y1000	Y1000	Y1000	Y1000	K0000	Y0000	86
Z	2LEZ7	2ZLE7	2ZEL3	2ZEL0	Z2300	Z0000	Z0000	43
Recognised Correctly	67	52	73	58	79	61	55	**63**

Table 3.2.4

Font Size Character	Veranda Bold 20pt	18pt	16pt	14pt	12pt	10pt	8pt	Recognised Correctly
0	0S6C9	QC09S	Q0S98	0SQC6	0C6S9	-	-	71
1	7Z000	32Z00	2Z300	32Z00	ZEL00	FZ000	1J000	14
2	32ZE7	2EFLZ	2Z37E	2FEZ0	2Z300	Z2000	Z2000	57
3	3JS00	3JS00	30000	30000	3J000	37000	30000	100
4	40000	40000	-	-	40000	40000	40000	71
5	5SEF0	5SEF0	5SFE0	5FSE0	5SFE0	5SFE0	FS5E0	86
6	5FSE0	60000	5FES0	5FES0	5FSE0	FES50	D0000	14
7	73000	73000	73000	73000	73T00	70000	70000	100
8	-	-	B0000	PRB00	-	D0000	-	0
9	SJ530	3JS50	Q9S03	5BD00	-	B0000	-	0
A	PBRD0	00000	BPR00	BPR00	40000	A0000	B0000	14
B	00000	PRB00	PBR00	BPDR0	PRB00	B0000	-	29
C	FEC00	Q9CS0	CQ09S	FEC00	FEC00	FEC00	BP000	14
D	DS600	DSPB0	DSPB6	DSP6B	DSPB0	DPR00	DPR00	100
E	EFL00	EFL00	EFL00	EF000	FE500	EFSL5	EFL00	86
F	FES50	FES50	FES50	FES50	FES50	FES50	FE5S0	100
G	GC000	C0000	CG000	CG000	CG000	G0000	BP000	29
H	H1000	H1000	H1000	H1000	H1000	H0000	H0000	100
J	3JS00	3J700	3JS00	3J000	3J000	37000	-	0
K	KX100	1KUV0	XK100	1VU00	10000	10000	-	14
L	L1E20	L1EJ0	L1E00	L1E00	L1E00	L1E00	L1000	100
M	MH000	1H000	M1000	M1000	M0000	M0000	00000	71
N	1YM00	XK100	10000	N1000	N1000	N0000	N0000	57
P	PRD00	PRD00	PR000	PRDB0	PRDB0	-	BP000	71
R	PRDB0	PBDR0	RP000	PRB00	BP000	-	BP000	14
S	S5F3E	S0600	SQ000	5SFE0	S5300	S5FE0	BP000	71
T	T0000	T7F3S	T7000	T0000	T0000	T0000	T0000	100
U	U1JV0	U1JV0	U1J00	U1J00	U1J00	1UJ00	1UJ00	71
V	V1UJH	U1JV0	U1J00	1UJ00	1JU00	10000	WN000	14
W	N0000	WV000	1U000	NW000	NW000	NWKV0	10000	14
X	1X000	10000	XK000	XK000	10000	X0000	-	43
Y	Y0000	Y1000	Y1000	Y0000	1Y000	BP000	BP000	57
Z	2ZEL3	2Z73E	2ZEL7	2Z37E	Z273E	Z2000	Z2000	43
Recognised Correctly	58	52	55	55	52	58	39	**52**

Table 3.2.6

Font Size Character	EuroStile Bold 20pt	18pt	16pt	14pt	12pt	10pt	8pt	Recognised Correctly
0	Q09S8	0QSC9	QS908	C0G00	Q98C0	-	DPRS0	29
1	10000	-	70000	10000	10000	-	-	43
2	2DELS	2ELZ0	2QELS	2LEZF	2QELF	Z2EF0	Z0000	71
3	SQ000	Q0000	Q9S80	3JS50	-	50000	37J00	29
4	00000	00000	40000	40000	-	00000	40000	43
5	5SEF0	5FES0	5SEF0	5ESF0	5ESF0	5FSE0	-	86
6	5FS6E	SQ6D0	6DS00	PRB00	00000	EF000	E5F00	14
7	73000	73000	73000	73000	73000	70000	7T000	100
8	BPD00	Q9800	B9000	Q9800	Q9000	DB000	-	0
9	53JDS	Q98S0	Q9800	Q9800	Q0000	00000	-	0
A	DB000	40000	B0000	00000	-	00000	-	0
B	PBR00	PBRD0	B0000	BD000	D0000	B0000	-	43
C	C0000	C0000	QC000	CQ000	2EFL0	FESL5	B0000	43
D	DBSPQ	DSBP0	DB000	DSQ00	DQ000	-	DP000	86
E	EFL00	EFL00	EFL00	EFL00	EFL2Z	EF000	EF500	100
F	FE5S0	FES50	FES50	FES00	FE000	FES50	FE500	100
G	C0000	5FS00	G0000	QC09G	Q98C0	FE5S0	FE5S0	14
H	H1000	H1000	H1000	H1000	H1000	H0000	H0000	100
J	1UJ00	1JU00	1JU00	1JU00	J1000	J1000	10000	29
K	1U000	XFK1E	1LE00	X1K00	K1X00	K0000	1J000	29
L	L1E2J	L1E20	L1E20	L1E2Z	L1E00	LE100	L0000	100
M	M0000	M0000	N0000	P0000	00000	M4000	BP000	43
N	J1000	1JN00	N1000	NWV00	NWV00	N0000	N0000	71
P	PRDBS	PRDB0	PRDBS	PRB00	PRD00	00000	BP000	71
R	PRB00	RP000	PRB00	B0000	B0000	B0000	-	14
S	F5S00	SQD60	JS350	S5F00	3JS00	E5F00	FE5S0	29
T	T0000	T7000	T7000	T0000	T0000	T0000	T0000	100
U	U1J00	UJ1V0	1UJV0	1UJV0	1UJ00	1UJ00	1UJ00	29
V	U1JV0	1UJVH	1UJV0	VW1N0	WVNK0	1J000	NK000	14
W	WV000	K0000	WV000	1VU00	WV000	-	10000	43
X	-	XK000	XK000	XK000	X0000	-	W0000	57
Y	Y1000	Y1000	Y1000	Y1000	Y0000	Y0000	-	86
Z	2ZEL7	2ZLE7	2ZEL0	Z2LE0	Z2000	Z2000	Z0000	57
recognized Correctly	48	55	55	64	55	45	33	**51**

Table 3.2.7

Font Size Character	License Plate 22px	18px	16px	14px	12px	10px	Recognised Correctly
0	Q9C00	Q98S0	Q0CS9	Q09SD	DS6P0	D0000	0
1	1JU00	1JU00	1JU00	10000	1JU00	1J000	100
2	2ZL3E	23700	2Z300	-	2ZL00	32700	67
3	3JS00	J0000	-	3J000	3J000	37000	67
4	40000	41000	40000	40000	40000	1H000	83
5	5SEF0	5S3JF	5SE00	5S000	5S000	-	83
6	1J000	60SJ0	6S0C0	1J000	1J000	10000	33
7	73000	70000	70000	70000	73T00	70000	100
8	B0000	BP000	Q8000	Q0000	B0000	D0000	0
9	9QC00	9Q000	Q9C00	9Q000	D0000	-	50
B	PBR00	BDP00	PBDR0	BD000	B0000	-	50
D	DBPQS	DS000	DBPRS	DB000	SD60Q	DPR00	83
F	FE000	FE000	FE000	FE000	FE000	-	83
G	G0000	G0000	G0000	G0000	CG000	D0000	67
H	H1000	1HJ00	H1000	H1000	H1000	1H000	67
J	J3100	J3100	J1300	J1U00	J1000	J1000	100
K	1JKXU	KX000	K1000	K0000	K0000	K0000	83
L	L1EJU	L1EJU	L1EJ0	L1JEU	L1EJ0	LJ1E0	100
N	N1000	N1000	N1000	N0000	N0000	1J000	83
P	PRD00	PRD00	P0000	PRB00	PR000	P0000	100
R	RP000	RXK00	R0000	RP000	RP000	-	83
S	S3J50	3JS00	S53F0	S6D00	S53J0	50000	67
T	T7000	M1000	YT000	Y0000	1J000	1J000	17
V	1YVUJ	1VYK0	1VK00	VWN00	NKV00	KN000	17
X	XK000	KX1J0	KX000	K1V00	-	K0000	17
Y	Y1000	Y1000	10000	Y0000	Y0000	KN100	67
Z	Z27LE	Z2L7E	Z2LE0	2ZLE3	ZL2E0	ZLE20	83
Recognised Correctly	78	70	67	74	67	33	**56**

Table 3.2.8.

character input	recognized as 0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	G	H	J	K	L	M	N	P	Q	R	S	T	U	V	W	X	Y	Z	?	!	#	Rates in %: classified correctly	mis Class.	un classified
0	11											1	1	11											19										5	32	48	22.9	66.7	10.4
1		15	1	2				3								1		4			1	5												2	14	19	48	31.3	39.6	29.2
2			35	2											2																			8	1	12	48	72.9	25.0	2.1
3	1			27		1						3							1						2		1								12	9	48	56.3	18.8	25.0
4	3	1			34																														10	4	48	70.8	8.3	20.8
5						37									2	3											2								4	7	48	77.1	14.6	8.3
6	3	4				11	12					3		1	2	3								3	2		2								2	34	48	25.0	70.8	4.2
7								48																											0	0	48	100.0	0.0	0.0
8	3			1					8	1		13	1	3										1	8										9	31	48	16.7	64.6	18.8
9	1	1		1		3				7		5		1										1	17		1								10	31	48	14.6	64.6	20.8
A	3				6						17	6		1										2											7	18	42	40.5	42.9	16.7
B	3											21		8										9		2									5	22	48	43.8	45.8	10.4
C	1		4									2	15	1	1	12			2						4										0	27	42	35.7	64.3	0.0
D	1													43											2		1								1	4	48	89.6	8.3	2.1
E															40	2																			0	2	42	95.2	4.8	0.0
F	1															46																			1	1	48	95.8	2.1	2.1
G	2			1		1						1	14	2		5	10								8		1								3	35	48	20.8	72.9	6.3
H	1	2																45																	0	3	48	93.8	6.3	0.0
J		18		9															18				1						1						1	29	48	37.5	60.4	2.1
K		16																		20												11			1	27	48	41.7	56.3	2.1
L																					48														0	0	48	100.0	0.0	0.0
M	5	1										2						1				19	1	7							2		1		3	20	42	45.2	47.6	7.1
N		19																	1				26									1			1	21	48	54.2	43.8	2.1
P	1											3												43											1	4	48	89.6	8.3	2.1
Q																																								
R												10												9	1	22						2			4	22	48	45.8	45.8	8.3
S				4		4						4	1	2	1	2			1						1		26								2	20	48	54.2	41.7	4.2
T		2																				1						43					2		0	5	48	89.6	10.4	0.0
U		17																											25						0	17	42	59.5	40.5	0.0
V		16																	1	5			3						4	15	4				0	33	48	31.3	68.8	0.0
W		8																1		1			11						1		15				5	22	42	35.7	52.4	11.9
X		5																		9											1	28			5	15	48	58.3	31.3	10.4
Y		2										2								4													38		2	8	48	79.2	16.7	4.2
Z			21																															27	0	21	48	56.3	43.8	0.0
																																						57	36	7

?: number of characters unclassified
!: number of characters misclassified
#: total number of characters

Table 3.2.9. Same as table 3.2.7 only the neocognitron specified in section 2.3 of this appendix have been used

Font Size / Character	License Plate 22px	18px	16px	14px	12px	10px	Recognised Correctly
0	-	09000	0G800	00000	D0000	D0000	50
1	1J000	1J000	1J000	T0000	1J000	10000	83
2	23Z7L	273Z0	2Z370	-	20000	30000	67
3	30000	J3000	-	3J000	3J500	37000	67
4	40000	40000	-	41000	40000	H1000	67
5	5SF00	5SF30	5SF00	5S000	5S000	-	83
6	60000	06000	60000	60000	6V000	V0000	67
7	73T00	7T000	70000	7T000	73T00	70000	100
8	B0000	B0000	-	90000	B0000	D0000	0
9	90000	90000	-	90000	-	Y0000	50
B	B0000	B0000	-	B0000	-	-	50
D	D0000	D0000	DP000	D0000	D0000	DP000	100
F	FS000	F0000	F0000	FL000	FL2S0	FL200	100
G	G0000	G0000	-	G0000	G0000	D0000	67
H	H1000	1H000	H1000	H0000	H0000	1H000	67
J	J3100	J3100	J1300	J1000	J1000	J1000	100
K	1K000	KX000	1K000	K0000	K0000	K0000	67
L	L1200	L1200	L1200	L1200	L1200	L2000	100
N	N1000	N1000	N1000	N0000	N0000	10000	83
P	PD000	PD000	P0000	PR000	P0000	P0000	100
R	R0000	RXK00	R0000	R0000	R0000	-	83
S	3J500	3J000	S5F30	5JSD0	5S000	50000	17
T	T0000	14000	T0000	YT000	1J000	10000	33
V	1Y000	1Y000	1Y000	10000	1J000	10000	0
X	X0000	1XK00	XK100	1K000	-	K0000	33
Y	Y0000	Y1000	1YH00	Y0000	Y0000	Y0000	83
Z	2ZL73	2ZL70	2ZL00	2ZL30	ZL200	ZL200	33
Recognised Correctly	77.78	66.67	62.96	70.37	70.37	40.74	**56**

3.3 Three layered network based on thinned character samples of normalised license plate character cut-outs

The tables in this section contains the recognition results on the network defined in section 2.4 of this appendix. The input samples are binary images of characters with the license plate font only. This font is shown in figure 2.9 of the main document.

Table 3.3.1.

Font Size / Character	License Plate 22px	18px	16px	14px	12px	10px	Recognised Correctly
0	0DG83	0D8G0	0G830	0D900	-	40000	66.7
1	1YVN0	1YVN0	1YVN0	T0000	10000	10000	83.3
2	2Z000	2Z000	2Z000	-	2LZ00	-	66.7
3	3J000	3J000	3J000	3J000	30000	70000	83.3
4	40000	40000	40000	40000	40000	40000	100.0
5	5B000	5S000	5S000	50000	50000	-	83.3
6	60000	60000	60000	60000	-	40000	66.7
7	7TZ00	7TZ00	70000	7T000	70000	70000	100.0
8	B8G30	B8D60	B8000	B9000	-	-	0.0
9	90000	90000	90000	94000	40000	-	66.7
B	B8DP0	BD8PR	BDP80	B8P60	B8600	B0000	100.0
D	D0B8G	DB000	DB000	DB200	D0000	D0000	100.0
F	FP500	FP000	FP000	FP000	F0000	F0000	100.0
G	G0000	G0000	G0000	G0000	-	-	66.7
H	HKN00	HNKV0	HKN00	HK000	HK000	HK000	100.0
J	J3000	J3000	J3000	J0000	J0000	J0000	100.0
K	HK1N0	KN000	KNV00	KN000	KN000	K0000	83.3
L	L2000	L2000	L2000	L0000	L0000	L0000	100.0
N	NVKH0	NVHKX	NVKH0	NV000	NK000	NK000	100.0
P	PR000	PR000	PR000	PR000	PR000	-	83.3
R	RP000	RP000	RP000	RP800	RP000	-	83.3
S	S0000	SG000	S5000	S0000	50000	-	66.7
T	T7000	10000	T0000	T0000	10000	10000	50.0
V	VNYX0	VNY1X	VNY00	VN000	VK000	-	83.3
X	XY000	YX000	XYKNV	YXVNK	XY000	KY000	50.0
Y	YVX10	YVX10	YVXN0	YVXK0	YV000	YVK00	100.0
Z	Z2700	Z2700	Z2000	Z2000	Z2L00	Z2L00	100.0
Recognised Correctly	93	89	96	85	74	48	**81**

Table 3.3.2.

Font Size / Character	License Plate 22px	18px	16px	14px	12px	10px	Recognised Correctly
0	0DG83	0D8G0	0D8GB	0D8G0	G0000	0D8BG	83.3
1	1YVN0	1YVN0	1VY00	1YVN0	1NVYH	1VNY0	100.0
2	2Z000	2Z000	2Z000	-	2ZL00	20000	83.3
3	3J000	3J000	3B800	3J000	37000	3B000	100.0
4	40000	40000	40000	40000	40000	40000	100.0
5	5B000	5S000	5SG00	50000	5S000	5S000	100.0
6	60000	60000	60000	60000	60000	60000	100.0
7	7TZ00	7TZ00	7T000	7T000	7T000	7T000	100.0
8	B8G30	B8D60	B8D39	8BP93	B8600	8BD00	33.3
9	90000	90000	98000	90000	90000	93800	100.0
B	B8DP0	BD8PR	B86PD	B8D6P	BD8P0	B86PD	100.0
D	D0B8G	DB000	D0B3G	DB020	DB000	0DG00	83.3
F	FP500	FP000	F0000	FP000	FP000	F0000	100.0
G	G0000	G0000	G0000	G0000	-	-	66.7
H	HKN00	HNKV0	HKN00	HKN00	HKN00	HK000	100.0
J	J3000	J3000	J3000	J3000	J3000	J3000	100.0
K	HK1N0	KN000	XKY00	KXN00	KNV00	K0000	66.7
L	L2000	L2000	L0000	L2Z00	L0000	L0000	100.0
N	NVKH0	NVHKX	HKX00	NVKHX	NHVK0	HKNXY	66.7
P	PR000	PR000	PRF80	PR000	PRF8B	PRFB8	100.0
R	RP000	RP000	-	RP000	RPBF0	RP000	83.3
S	S0000	SG000	S5000	5S000	S5000	S5000	83.3
T	T7000	10000	T0000	T0000	1YVN0	1VNHY	50.0
V	VNYX0	VNY1X	VNY00	-	VN000	VN000	83.3
X	XY000	YX000	YXNV0	VYXNK	XY000	XYKNV	50.0
Y	YVX10	YVX10	YVX1N	YVXNH	Y1VX0	YVX1N	100.0
Z	Z2700	Z2700	Z2L00	Z2700	Z2L00	Z2L00	100.0
Recognised Correctly	93	89	81	85	85	85	**86**

Table 3.3.3

Font Size Character	License Plate 22px	18px	16px	14px	12px	10px	Recognised Correctly
0	0DG83	0D8G0	0G830	0D900	G0000	0D8BG	83.3
1	1YVN0	1YVN0	1YVN0	1YVN0	1NVYH	1VNY0	100.0
2	2Z000	2Z000	2Z000	-	2ZL00	20000	83.3
3	3J000	3J000	3J000	3J000	37000	3B000	100.0
4	40000	40000	40000	40000	40000	40000	100.0
5	5B000	5S000	5S000	50000	5S000	5S000	100.0
6	60000	60000	60000	60000	60000	60000	100.0
7	7TZ00	7TZ00	70000	7T000	7T000	7T000	100.0
8	B8G30	B8D60	B8000	B9000	B8600	8BD00	16.7
9	90000	90000	90000	94000	90000	93800	100.0
B	B8DP0	BD8PR	BDP80	B8P60	BD8P0	B86PD	100.0
D	D0B8G	DB000	DB000	DB200	DB000	0DG00	83.3
F	FP500	FP000	FP000	FP000	FP000	F0000	100.0
G	G0000	G0000	G0000	G0000	-	-	66.7
H	HKN00	HNKV0	HKN00	HK000	HKN00	HK000	100.0
J	J3000	J3000	J3000	J0000	J3000	J3000	100.0
K	HK1N0	KN000	KNV00	KN000	KNV00	K0000	83.3
L	L2000	L2000	L2000	L0000	L0000	L0000	100.0
N	NVKH0	NVHKX	NVKH0	NV000	NHVK0	HKNXY	83.3
P	PR000	PR000	PR000	PR000	PRF8B	PRFB8	100.0
R	RP000	RP000	RP000	RP800	RPBF0	RP000	100.0
S	S0000	SG000	S5000	S0000	S5000	S5000	100.0
T	T7000	10000	T0000	T0000	1YVN0	1VNHY	50.0
V	VNYX0	VNY1X	VNY00	VN000	VN000	VN000	100.0
X	XY000	YX000	XYKNV	YXVNK	XY000	XYKNV	66.7
Y	YVX10	YVX10	YVXN0	YVXK0	Y1VX0	YVX1N	100.0
Z	Z2700	Z2700	Z2000	Z2000	Z2L00	Z2L00	100.0
Recognised Correctly	93	89	96	89	85	85	**90**

3.4 Verification and evaluation results on real–world photographs

Table 3.4.1 in this section contains the recognition results when using the network defined in section 2.4 combined with the networks defined in 2.5. Table 3.4.2 in this section shows the recognition results when using the network defined in 2.6 of this appendix.

Table 3.4.1 Photographs used in verification testing. The images are printed in appendix IV. Refer section 4.4 of the main document.

ref	**image #**	**plate ident**	**char size in px**	**Network configuration**									
				fukisuma	**cornet 3**	**cornet 4**	**cornet 15**	**cornet 17**	**d/c**	**reg**	**err**	**rej**	**cornet 15+ >9**
1	004	**02-DR-GR**	12	Q?D9?9	Q2D9GQ	02D9G?	02DRGR	02DRGR		6	0	0	02DRGR
2	005	**02-DR-GR**	12	9LDR4P	02DRGP	02DRGP	02DRGR	02DRGR		6	0	0	02DRGR
3	006	**02-DR-GR**	11	CZD9?R	02D9FR	02D?FR	02DRGR	02DRGR		6	0	0	02DRGR
4	007	**02-DR-GR**	11	C2DR??	C2DR1X	02DR?X	02DRGR	02DRGR		6	0	0	02DRGR
5	008	**02-DR-GR**	13	02SRCR	02DRCR	02DRGR	02DRGR	02DRGR		6	0	0	02DRGR
6	009	**02-DR-GR**	13	0LD?CR	02D?GR	02D?GR	020RGR	02DRGR		6	0	0	02DRGR
7	012	**02-DR-GR**	16	QLDR?R	Q2DRFR	02DXFR	02DRGR	02DRGR		6	0	0	02DRGR
8	013	**LS-BZ-92**	17	L5DL?2	LSDZ?2	LSBZ92	LSBZ92	LSBZ92		6	0	0	LSBZ92
9	015	**PG-ZG-50**	14	RCL?E0	PGZF50	PGZF50	PGZG50	PGZG50		6	0	0	PGZG50
10	016	**HJ-XT-50**	10	H4KFE2	H1X?5Q	11XF50	HJXT50	HJXT50		6	0	0	HJXT50
11	021e	**71-FF-NN**	16	71FFNN	71FFNN	71FFNN	71FFNN	71FFNN		6	0	0	71FFNN
12	022	**71-FF-NN**	14	7K?FNN	7XFFNN	71FFNN	7?FFNN	7?FFNN		5	0	1	7?FFNN
13	023	**RS-ZF-23**	15	R?LF2?	RS2F23	RS2F23	RSZF23	RSZF23		6	0	0	RSZF23
14	024	**68-DB-BT**	14	??DDQT	6?DDDT	6B?DBT	6BDBBT	6BDBBT	8/B	5	1	0	6BDBBT
15	028	**PD-21-DJ**	12	RSZ1S1	PQ31D1	P?3101	P021DJ	PD21DJ		6	0	0	PD21DJ
16	029	**TF-TF-85**	12	TFP?D5	TFP?B?	TFPFD5	T??F?5	T??F?5		3	0	3	T??F?5
17	031e	**NX-NP-31**	9	4?P??1	1K1Y?1	1K?Y?1	NXNP21	NXNP21		5	1	0	N/A
18	032	**VL-29-DN**	9	VLL?1?	KLL?1?	?LL11J	?LZ9DN	?LZ9DN	2/Z	4	1	1	N/A
19	035	**JS-PT-84**	9	4FP7A?	15P???	15P1??	JSPT84	J5PT84	S/5	5	1	0	N/A
20	036	**VG-47-GD**	17	44474S	1GH7GD	1GH7G0	VG47G0	VG47GD		6	0	0	VG47GD
21	037e	**LT-TL-53**	20	L?TL5?	LXTL53	LXFL53	LTTL53	LTTL53		6	0	0	LTTL53
22	038	**HP-BG-77**	15	49DC77	HQDG77	10BG77	HPBG77	HPBG77		6	0	0	HPBG77
23	039	**PX-HT-87**	17	PKHTQ7	PXHTQ7	PXHT07	PXHT07	PXHT07		5	1	0	PXHT07
24	040e	**VR-18-NK**	9	_4PK1?	1PK1?K	_??11?	VR18?K	VR18?K		5	0	1	N/A
25	041	**70-VH-RT**	10	7QYHRT	7QYHRT	701HRF	70VHRT	70VHRT		6	0	0	70VHRT
26	043e	**HN-LX-85**	17	H1L4Q5	H1LX8S	HNLX85	HNLXBS	HNLXB5	8/B	5	1	0	HNLXB5
27	044	**JG-XG-98**	19	14??93	1GXFQ8	1GXF98	1GXG90	JGXG90		5	0	1	JGXG90
28	045e	**KZ-96-XR**	15	1296KR	1296XR	1296XR	KZ96XR	KZ96XR		6	0	0	KZ96XR
29	046	**?P-44-RD**	13	_P??RD	_P44RD	_P4?RD	_P44RD	_P44RD		5	0	0	_P44RD
30	047	**VS-44-ZT**	12	V???ZF	K3??ZF	1S??2F	YS44ZT	YS44ZT		5	1	0	YS44ZT
31	048	**BB-PB-95**	14	D?SB95	BBPBQ5	BBPD95	BBRB95	BBRB95		5	1	0	BBRB95
32	049e	**BB-NH-08**	18	DBKH0?	P?KH0?	P?NH0B	BBNH08	BBNH08		6	0	0	BBNH08

ref	image #	plate ident	char size in px	Network configuration fukisuma	cornet 3	cornet 4	cornet 15	cornet 17	d/c	reg	err	rej	cornet 17 > 9
35	053	**ZF-54-FY**	14	LF5?FY	2F54FY	2F54FY	ZF54FY	ZF54FY		6	0	0	ZF54FY
36	054	**50-VG-ST**	19	5SV45T	5SVGST	501G5T	50VGST	50VGST		6	0	0	50VGST
37	055e	**JX-71-RT**	15	1111R7	1X11R7	1X11X7	JXH1RT	JXH1RT		5	1	0	JXH1RT
38	056	**SJ-GN-95**	11	?W4ND5	51?NB5	51?N?5	SJGN9S	SJGN95		6	0	0	SJGN95
39	057e	**HZ-LH-47**	12	4L1457	1Z11?7	N211?7	NZLH47	NZLH47		6	0	0	NZLH47
40	058	**XN-ZD-18**	13	?KLI16	X1ZD16	XXZD10	XKZD1B	XKZD18		5	1	0	XKZD18
41	059	**PN-NG-01**	14	PN1C01	PN1G01	PN?G01	PNNG01	PNNG01		6	0	0	PNNG01
42	060	**RS-BJ-46**	13	R5D146	RSD146	RSD146	RSBJ46	RSBJ46		6	0	0	RSBJ46
43	061	**XJ-PJ-42**	16	K1R142	X1RJ42	X1PJ42	X1PJ42	XJPJ42		6	0	0	XJPJ42
	character recognition rate			38.91	60.70	63.81	89.88	92.61		92.6	4.28	3.11	93.99
	plate recognition rate			2.33	4.65	11.63	53.49	65.12					71.79
	plate recognition rate theoratically			0.35	5.00	6.75	52.73	63.08					68.95

Table 3.4.2 Photographs used in for system evaluation refer section 6.1.2 and 6.1.3 of the main document.

ref	image #	plate ident	char size pixels	#seg	met	plate read	final system output string considering all segments locations of segments not shown here
1	001	**ST98BG**	12	10	1	**ST98BG**	[?L??ST98BG]
2	002	**GHBB86**	13	27	2	**GHBB86**	[-?1?1???V-GHBB86?1?]
3	003	**NTPJ54**	14	10	1	**NTPJ54**	[L??1NTPJ54]
4	004	**NVDV26**	12	12	3	**NYDN26**	[-?NYDN26]
5	005	**PP08HZ**	12	14	3	**PP08HZ**	[????L?1?PP08HZ]
6	006	**XPDR13**	14	16	1	**XPDD13**	[-1????XPDD13]
7	007	**SZSR76**	17	8	1	**SZSR76**	[1?SZSR76]
8	008	**RXFD02**	13	6	1	**RYFD02**	[RYFD02]
9	009	**XSFT38**	16	16	2	**XSFT38**	[1-????XSFT38]
10	010	**PFVH28**	15	9	1	**PFVH28**	[?H28?PFV?]
11	011	**LPNF11**	15	9	1	**LPNF11**	[???LPNF11]
12	012	**02DRGR**	13	9	1	**02DRGR**	[??102DRGR]
13	013	**ZG27RY**	14	22	1	**ZG27HY**	[?1-??L1???ZG27HYH?]
14	014	**DBXN69**	14	9	1	**DB?H69**	[?DB?H69?L]
15	016	**YG90DH**	12	14	2	**YG90DH**	[????11?YG90DH?]
16	017	**PZ11VS**	11	11	3	**PZ11VS**	[??11PZ11VS?]
17	018	**VTBD56**	11	27	1	**VTBD56**	[-????LK?K????1?11VTBD56]
18	020	**YR76PH**	10	12	3	**?R76PH**	[?11????R76PH]
19	021	**90DZLX**	13	13	2	**90DLLX**	[11-90DLLX]
20	022	**JBFH95**	11	12	3	**JBFH95**	[???1??JBFH95]
21	023	**LFFN14**	12	6	2	**LFFH14**	[LFFH14]
22	024	**97FBFN**	12	21	2	**97F?FN**	[????11?11???1?97F?FN?]
23	025	**ZF77VR**	9	39	2	**ZF77VR**	[-???1????1-????97FBFN??ZF77VR1?]
24	026	**JRFP34**	11	14	1	**JRFP34**	[???11J??JPFP34]
25	027	**LK63VV**	12	12	1	**LR63YV**	[???1?1LR63YV]
26	028	**FXBB03**	12	7	3	**FXBB03**	[FXBB03?]
27	029	**YL87KD**	12	9	1	**YL87KD**	[???YL87KD]
28	030	**RSDL51**	11	12	3	**RSDL51**	[???1?JRSDL51]
29	033	**86DNZT**	15	9	1	**86DNZT**	[?86DNZTNL]

ref	image #	plate ident	char size pixels	#seg	met	plate read	final system output string considering all segments locations of segments not shown here
32	036	**DXTR60**	11	16	1	**DXTR6D**	[?71116DDXTR1??L?]
33	037	**NNZB41**	11	18	1	**NNZB41**	[?1NNZB411??K-?]
34	038	**XN41YS**	11	9	1	**XN41YS**	[??LXN41YS]
35	039	**SF34KD**	13	10	2	**SF04KD**	[?SF04KD?4?]
36	044	**65DJSN**	11	17	3	**65DJSN**	[?L??11??11165DJSN]
37	050	**SRDJ27**	12	8	1	**SRBJ27**	[?1SRBJ27]
38	051	**VR77JR**	13	8	1	**YR77?R**	[??YR77?R]
39	052	**DGXG92**	12	8	1	**DGXG92**	[?1DGXG92]
40	053	**HZVT86**	13	16	3	**HZVT86**	[1?1??1????HZVT86]
41	054	**24DTVN**	13	12	1	**24DTVN**	[11?1??24DTVN]
42	055	**TY32DY**	15	17	1	**TY32DY**	[??11??1????TY32DY]
43	056	**HVRH38**	11	43	2	**HVRH38**	[-?1????1K???K-?5????1111-HVRH38]
44	057	**LXXH96**	14	15	3	**LXXH96**	[-7L11LXXH96]
45	058	**GBPL68**	13	11	3	**?BPL6?**	[?L????BPL6?]
46	059	**TH15XG**	14	6	3	**TH15XG**	[TH15XG]
47	061	**90FNTP**	12	6	1	**9DFNTP**	[9DFNTP]
48	062	**PBRB26**	12	10	3	**PBRB26**	[?1??PBRB26]
49	064	**RTGH72**	15	6	1	**RTGH72**	[RTGH72]
50	065	**SK58LD**	12	7	1	**SK58LD**	[SK58LD?]
51	066	**YV88XR**	12	7	1	**YV88XR**	[?YV88XR]
52	067	**TBPN68**	16	6	1	**?8PN68**	[?8PN68]
53	068	**YL68DR**	11	9	3	**YL68DR**	[???YL68DR]
54	069	**SL75SF**	11	6	1	**SL75SF**	[SL75SF]
55	070	**27GFNS**	13	8	1	**Z7GFNS**	[??Z7GFNS]
56	071	**SRGG06**	13	9	1	**SRGG06**	[???SRGG06]
57	072	**TS43LD**	11	11	1	**TS43LD**	[1???1TS43LD]
58	073	**TY88DY**	14	8	1	**TY?8DY**	[??TY?8DY]
59	074	**JSBX36**	13	10	1	**JSBX36**	[LJSBX36???]
60	075	**44DBPH**	13	7	3	**44DBPH**	[?44DBPH]
61	077	**06GBKJ**	13	14	1	**06GBKJ**	[1???1?06GB?KJ?]

ref	image #	plate ident	char size pixels	#seg	met	plate read	final system output string considering all segments locations of segments not shown here
64	085	**99FLTG**	11	8	1	**99FLTG**	[1?99FLTG]
65	086	**GRVR76**	12	10	2	**GRVR76**	[1??GRVR76?]
66	087	**PRDN31**	13	14	1	**PRDN31**	[??1?1???PRDN31]
67	088	**XDNX57**	12	10	2	**XDNXS7**	[????XDNXS7]
68	089	**LZLX06**	11	9	1	**?ZLXD6**	[????ZLXD6]
69	090	**FJTX23**	10	10	3	**FJTX23**	[3L20FJTX23]
70	091	**JPFF33**	11	10	3	**JPFF33**	[JPFF33??FJ]
71	092	**10FVHZ**	11	10	2	**10FVHZ**	[??10FVHZ1?]
72	093	**RDZG39**	13	6	1	**RDZG39**	[RDZG39]
73	094	**NFFG98**	13	10	1	**NFFG98**	[11??NFFG98]
74	095	**TY66GD**	12	8	1	**TY66GD**	[??TY66GD]
75	097	**45DHGH**	9	26	2	**?5DHGH**	[????N?1?1?F???GF94HT?5DHGH]
76	099	**GHZH03**	11	8	3	**GHZH03**	[?GHZH03?]
77	101	**FBBZ81**	10	11	3	**FBBZ81**	[???FBBZ81??]
78	103	**RV31SK**	10	24	2	**RV31SK**	[??L???1-L?NLJ?RV31SK]
79	104	**TTDD55**	13	10	2	**TTDD55**	[?7???TTDD55]
80	105	**DBPH32**	9	6	2	**D8PH32**	[D8PH32]
81	106	**HRZB38**	12	23	2	**HRZB38**	[??LK--K??HRZB38]
82	108	**52FZFD**	11	14	1	**52FZFD**	[1?1????52FZFD?]
83	109	**12DGKV**	13	8	1	**12DGKV**	[12DGKV?1]
84	110	**ZL61RL**	10	18	1	**ZL61RL**	[????1N12DGKVZL61RL]
85	111	**SRDJ27**	15	9	1	**SRDJ27**	[???SRDJ27]
86	113	**ZD42JJ**	14	10	2	**ZD42JJ**	[????JZD42J]
87	114	**DHVN14**	14	23	1	**DHVN14**	[-???4DHVN1-????]
88	115	**ZFNB57**	16	7	1	**ZFNB57**	[157ZFNB]
89	116	**VV89DH**	15	7	3	**VV89DH**	[?VV89DH]
90	117	**NSLF23**	12	11	1	**NSLF23**	[J???1NSLF23]
91	118	**GLXS20**	12	11	3	**GLXSZD**	[Y????GLXSZD]
92	119	**FVFD65**	16	8	1	**FVFD65**	[?0FVFD65]
93	121	**ZN31GT**	13	7	1	**ZN31GT**	[ZN31GT?]

ref	image #	plate ident	char size pixels	#seg	met	plate read	final system output string considering all segments locations of segments not shown here
96	124	**ZNFS55**	16	9	2	**ZNFS55**	[?F?ZNFS55]
97	125	**TZ11LF**	10	7	1	**TZ11LF**	[?TZ11LF]
98	126	**SRLP15**	13	6	3	**SRLP15**	[SRLP15]
99	127	**YD55ZG**	13	13	1	**YD55ZG**	[-11YD55ZG]
100	128	**LJ49BP**	13	9	1	**LJ49BP**	[VF?LJ49BP]
101	130	**FLJR28**	12	9	3	**FLJR28**	[??FLJR28?]
102	133	**LZHJ39**	13	9	2	**LZHJ39**	[?1?LZHJ39]
103	134	**FNVB58**	13	10	1	**FNVB5B**	[?11?FNVB5B]
104	135	**XN34LV**	12	12	1	**XN34LV**	[???1N?XN34LV]
105	136	**NZDG05**	14	7	1	**NZDG05**	[?NZDG05]
106	137	**NVSB21**	16	7	2	**NVSB21**	[?NVSB21]
107	139	**XV37TL**	12	6	1	**XV37TL**	[XV37TL]
108	140	**52FFBT**	14	6	1	**5ZFFBT**	[5ZFFBT]
109	141	**XT43RD**	11	7	1	**YT?3RD**	[?YT?3RD]
110	142	**KR89KV**	11	13	1	**KR89?V**	[1???V?KR89?V?]
111	143	**HVRF95**	13	6	1	**HYRF95**	[HYRF95]
112	144	**ZJFG31**	12	6	1	**ZJFG31**	[ZJFG31]
113	145	**PZ94DL**	13	11	1	**PZ94DL**	[?1NPZ94DLHL]
114	146	**SH98SY**	11	11	1	**SH985Y**	[???SH985YNL]
115	147	**NNLD35**	9	10	1	**HNLD35**	[1?P?HNLD35]
116	148	**GPXT07**	11	8	1	**GPXT07**	[?GPXT07?]
117	149	**SGTH68**	11	6	1	**SGTH68**	[SGTH68]
118	151	**LHGH16**	13	6	1	**LHGH16**	[LHGH16]
119	153	**GDHG72**	13	12	2	**GDHG72**	[-GDHG72?]
120	154	**PNNG01**	12	6	1	**PHHG01**	[PHHG01]
121	155	**DSXB30**	12	7	1	**D5XB30**	[1D5XB30]
122	156	**PZRJ65**	12	6	1	**PZRJ65**	[PZRJ65]
123	157	**PDNR99**	14	6	1	**PDNR9?**	[PDNR9?]
124	161	**XHRN12**	14	15	2	**XHRN12**	[1????4111XHRN12]
125	162	**GLRR80**	11	8	1	**GLRR80**	[1?GLRR80]

recognition rate char level	**93.47 %**	**syntax forced recognition on plate level**	**79.20 %**
recognition rate plate level	**71.20 %**	**syntax forced recognition on plate level and leaving out all plates with char sizes < 10 pt**	**80.72 %**

Intentionally left blank

Appendix IV

Photographs used in verification testings

P 004 02-DR-GR

P 005 02-DR-GR

P 006 02-DRGR

P 007 02-DR-GR

P 008 02-DR-GR

P 009 02-DR-GR

P 012 02-DR-GR

P 013 LS-BZ-92

P 015 PG-ZG-50

P 016 HJ-XJ-50

P021e 71-FF-NN

P022 71-FF-NN

P 023 RS-ZF-23

P 024 68-DB-BT

P 028 PD-21-DJ

P 029 TF-TF-85

P 031e NX-ND-31

P 032 VL-29-DN

P 035 JS-PT-84

P 036 VD-47-GD

P 037e LT-TL-53

P 038 HP-BG-77

P 039 PX-HT-87

P 040e VR-18-NK

P 041 70-VH—RT
P 045e KZ-96-XR

P 043e HN-LX-31
P 046 VP-44-RD

P 044 JG-XG-98
P 047 VS-44-ZT

P 048 BB-PB-95

P 049e BB-NH-08

P 050 BG-ZG-21

P 051 VX-RX-10

P 053 ZF-54-FY

P 054 50-VG-ST

P 055e JX-71-RT

P 056 SJ-GN-95

P 057e HZ-LN-47

P 058 XN-ZD-18

P 059 PN-NG-01

P 060 RS-BJ-46

P 061 XJ-PJ-42