Data Classification using SVM & Quickcog

Mahesh Poolakkaparambil

Winter Semester 2007/08

Under the guidance of
Prof. Dr.-Ing. Andreas König
&
Kuncup Iswandy

Contents

Introduction

- Basic idea of this study project is to carry out the classification task and compare classification techniques K-NN, PNN & RNN with LS-SVM (Least Square Support Vector Machine) & SVM
- Data sets used are the Repository data from the website , http://www.ics.uci.edu/~mlearn/MLRepository.html and Robot vision data from Prof. Dr. Ing Andreas König
- The repository data used for this project are Breast cancer data, Segment data, Wine data and the Robot vision data (benchmark data)
- Entire experiment has been carried out on a Intel Celeron processor (1.73Ghz)

Introduction

- We used Quick-cog for the K-NN , PNN and RNN classifiers
- For SVM we used the LS-SVM MATLAB tool & SVM tool box
- Data division through out using Holdout (Source Mr.Iswandy)
- Seed used for division is ‘1‘ for all data
- 70% Training and 30% Testing (Division is proportional)
- We used normalized data through out our task(normalized btw -1 to +1).
- Experiments are carried out with different classifier parameters in a trial and error fashion to obtain better classification results (manual tuning)

Classification using LS-SVM

- In our experiment the Regularization parameter (Gamma) and Kernel parameter (Sigma) are varied btw 0.1 to 200 in different steps of various step size and tested each time for better results.
- Varied in steps of 1 or 5 till we get a good CR and then tuned carefully around the obtained values again with minute step size
- Values of these parameters with best classification rate are opted
- Encoding scheme used is code_MOC
- Implementation used is CMEX
- Kernel used is RBF
- Decoding schemes used is Hamming distance

LSSVM Classification

Data	# of Classes	# of Features	Total # of Samples	# of Training sets	# of Testing sets
Cancer	2	30	569	456	113
Segment	7	19	2310	1736	574
Wine	3	13	178	133	45

LSSVM Classification & Results

Data	Sigma	Gamma	Result (%)
Cancer	23.14	14.44	97.50
Segment	0.70	0.4	93.03
Wine	0.45	10	93.33

Other Classifiers (K-NN, PNN, RNN)

- Data sets are converted into .NIF format using MATLAB
- The classification methods are carried out without feature selection (Same as in LS-SVM)

Classifier parameters (K-NN, PNN, RNN)

K-NN

- Through out the Repository data we discussed the result for K=3 and K=5 for Robot vision data
- Distance and Radius values used were default (10000 , 1)

PNN

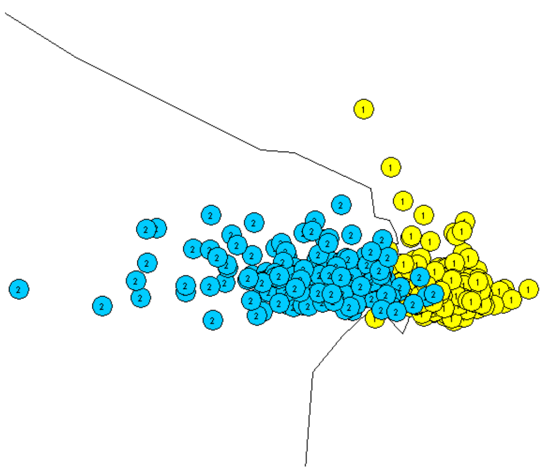
- Threshold and Sigma parameters are clearly mentioned with the Results
- We used Euclidean distance in PNN

RNN

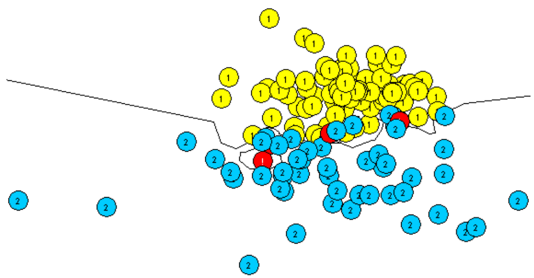
- We used Default values of Distance and radius as in case of K-NN
- We used Euclidean distance

K-NN(Cancer data)

Training



Testing(K=3)



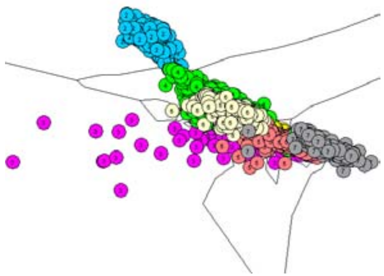
Confusion Matrix,1

Class1 (89):	0 (R)	95.500	4.490
Class2 (53):	0 (R)	3.770	96.220

Classification rate: 95.775 %

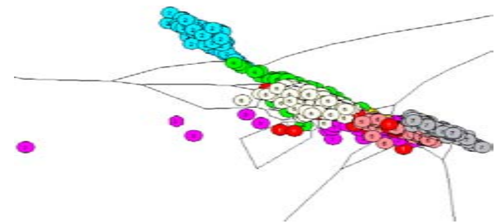
K-NN(Segment data)

Training



Testing(K=3)

Euclidean distance



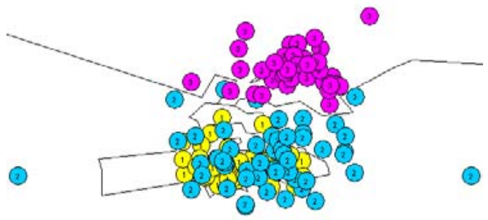
Confusion Matrix,1

Class1 (82):	0	(R)	98.780	0	0	0	0	1.210	0
Class2 (82):	0	(R)	0	100	0	0	0	0	0
Class3 (82):	0	(R)	0	0	0	89.020	0	10.970	0
Class4 (82):	0	(R)	0	0	0	6.090	92.680	1.210	0
Class5 (82):	0	(R)	0	0	0	2.430	4.870	92.680	0
Class6 (82):	0	(R)	0	0	0	0	0	0	100
Class7 (82):	0	(R)	0	0	98.780	0	1.210	0	0

Classification rate: 95.993 %

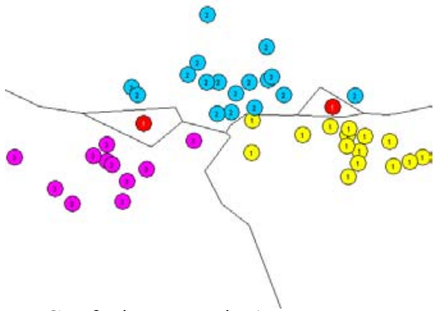
K-NN(Wine data)

Training



Testing(K=3)

Euclidean distance



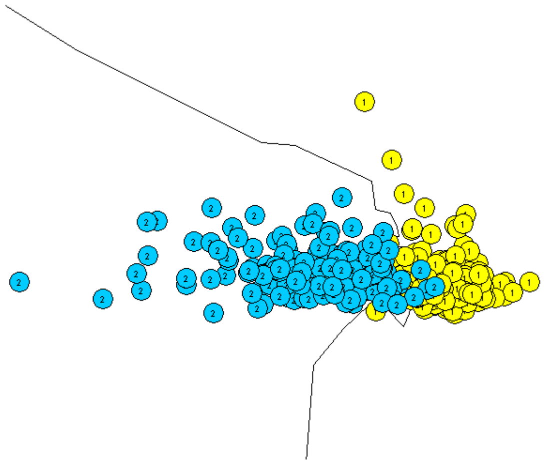
Confusions Matrix,1

Class1 (15):	0 (R)	100	0	0
Class2 (18):	0 (R)	5.550	94.440	0
Class3 (12):	0 (R)	0	0	100

Classification rate: 97.778 %

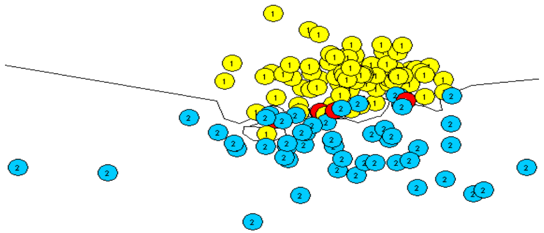
PNN(Cancer data)

Training



Testing

(Sigma=0.1, Threshold=1)



Confusions Matrix,1

Class1 (89):	0 (R)	96.620	3.370
Class2 (53):	0 (R)	3.770	96.220

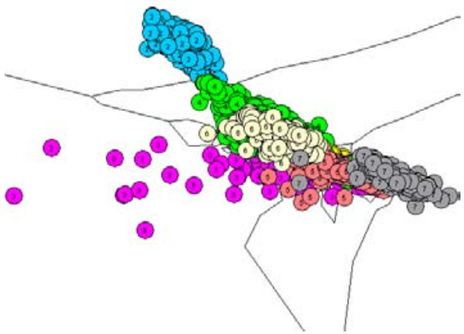
Classification rate: 96.479 %

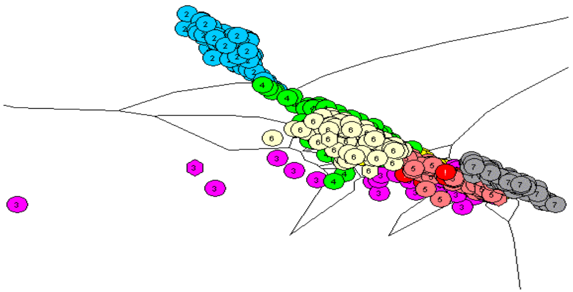
PNN(Segment data)

Training

Testing

(Sigma=0.1, threshold=1)





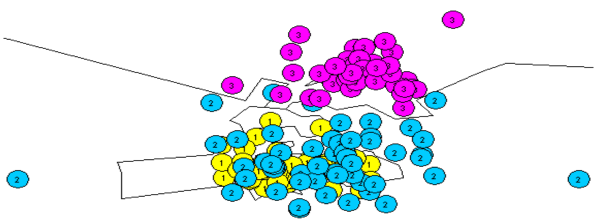
Confusion Matrix,1

Class1 (82):	0 (R)	98.780	0	0	0	1.210	0	0
Class2 (82):	0 (R)	0	100	0	0	0	0	0
Class3 (82):	0 (R)	0	0	91.460	0	8.530	0	0
Class4 (82):	0 (R)	2.430	0	2.430	89.020	6.090	0	0
Class5 (82):	0 (R)	1.210	0	8.530	2.430	87.800	0	0
Class6 (82):	0 (R)	0	0	0	0	0	100	0
Class7 (82):	0 (R)	0	0	0	1.210	0	0	98.78

Classification rate: 95.122 %

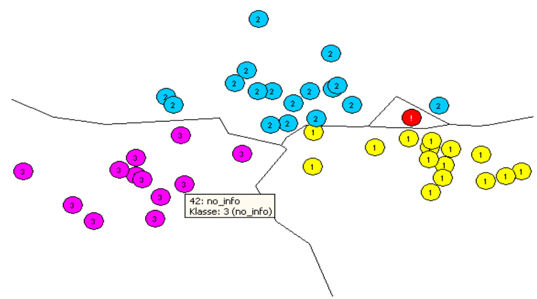
PNN(Wine data)

Training



Testing

(Sigma=0.5, Threshold=1)



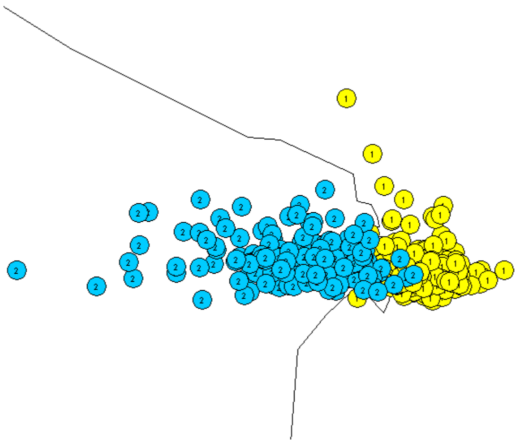
Confusion Matrix,1

Class1 (15):	0 (R)	100	0	0
Class2 (18):	0 (R)	5.550	94.44	0
Class3 (12):	0 (R)	0	0	100

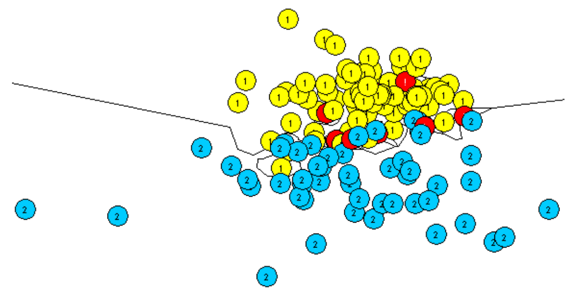
Classification srate: 97.778

RNN(Cancer data)

Training



Testing



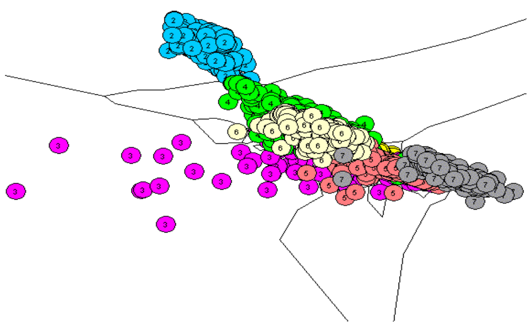
Confusion Matrix,1

Class1 (89):	0 (R)	93.250	6.740
Class2 (53):	0 (R)	3.770	96.220

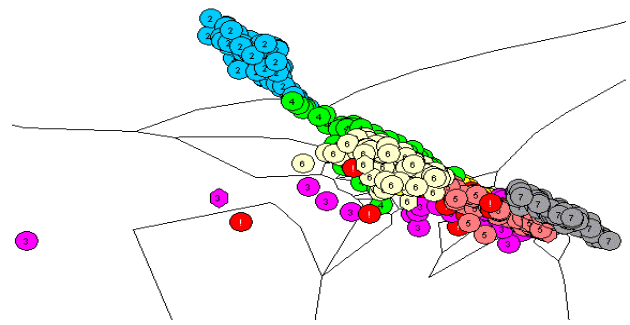
Classification rate: 94.366 %

RNN(Segment data)

Training



Testing



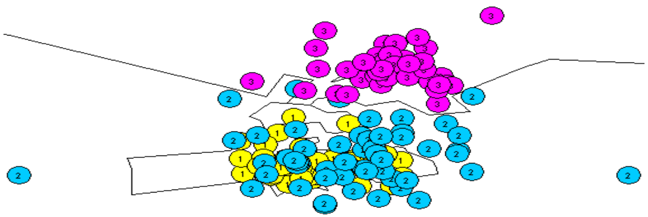
Confusion Matrix,1

Class1 (82):	0 (R)	98.780	0	0	0	1.210	0	0
Class2 (82):	0 (R)	0	100	0	0	0	0	0
Class3 (82):	0 (R)	0	0	82.920	2.430	14.630	0	0
Class4 (82):	0 (R)	0	0	4.870	91.460	1.210	2.430	0
Class5 (82):	0 (R)	1.210	0	2.430	6.090	90.240	0	0
Class6 (82):	0 (R)	0	0	0	1.210	0	98.780	0
Class7 (82):	0 (R)	0	0	0	1.210	0	0	98.7

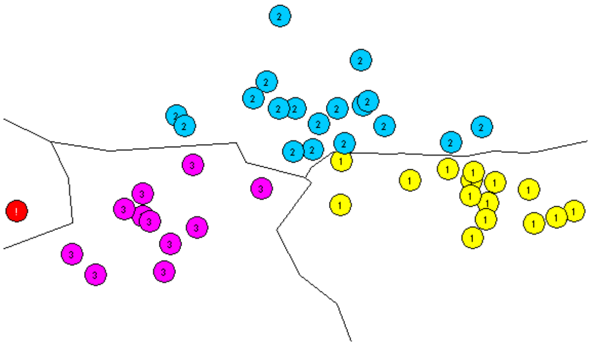
Classification rate: 94.425 %

RNN(Wine data)

Training



Testing



Confusion Matrix,1

Clase1 (15):	0 (R)	100	0	0
Clase2 (18):	0 (R)	0	100	0
Class3 (12):	0 (R)	0	8.330	91.660

Classification rate: 97.778 %

Classification Results(Comparison with other classifiers)

Data	K-NN (%)	PNN (%)	RNN (%)	LS-SVM (%)
Cancer	95.75	96.479	94.366	97.88
Segment	95.993	95.122	94.425	93.033
Wine	97.78	97.78	97.78	93.33

Classification of Robot vision

LS-SVM data preparations:

- Data is converted from .ewd format into .mat format using MATLAB
- Classes are re-ordered
- Normalization is same as before

Other classifier :

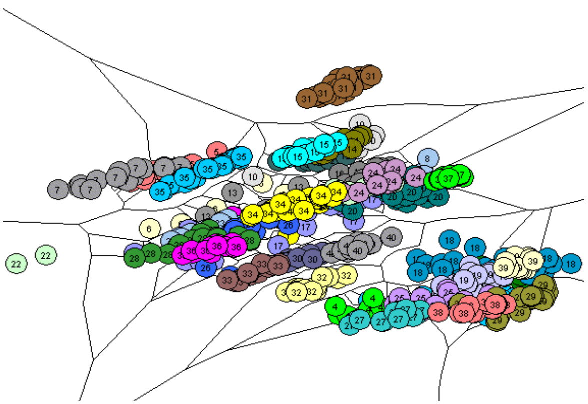
- Same as before & SVM

Classification data (Division and Features)

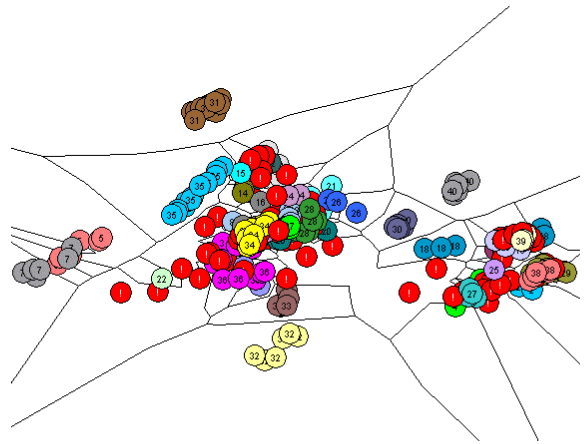
Data	# of data samples	# of classes	# of features	# of training	# of testing
Tiger	946	40	3	661	285
Tiger(top)	880	35	3	616	264
SPTH	945	37	3	663	282
Dorthrecht	260	11	3	182	78
VA	340	13	3	238	102
OLA	196	4	3	136	60

K-NN (Tiger cap set)

Training



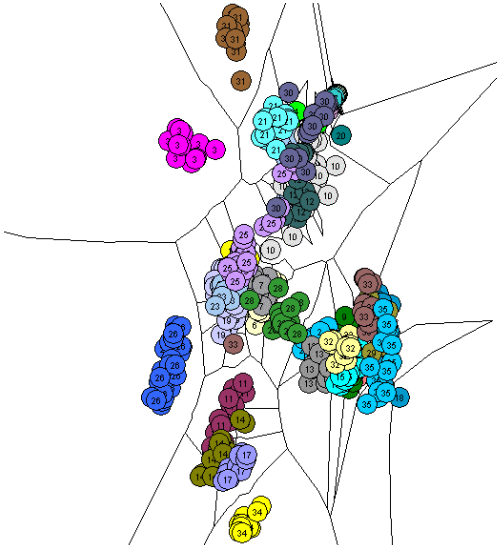
Testing



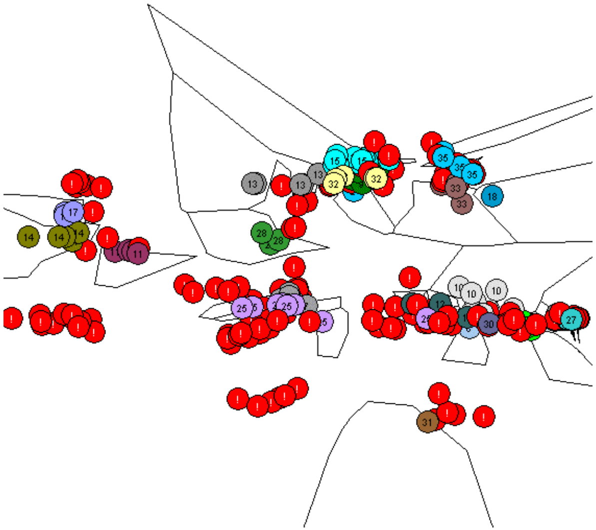
Classification rate: 67.368 %

KNN (Tiger cap set top only)

Training



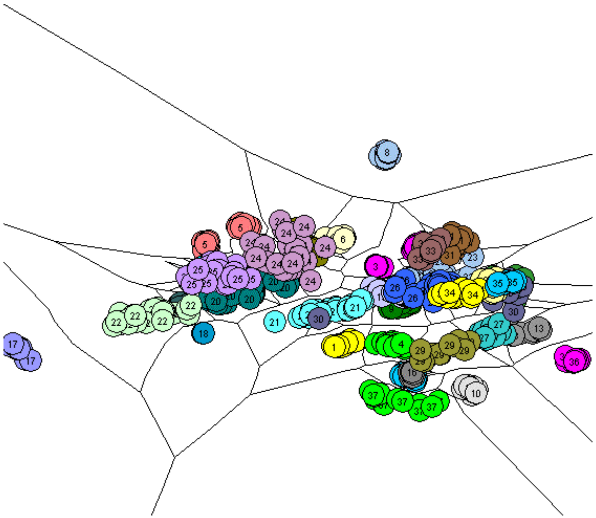
Testing



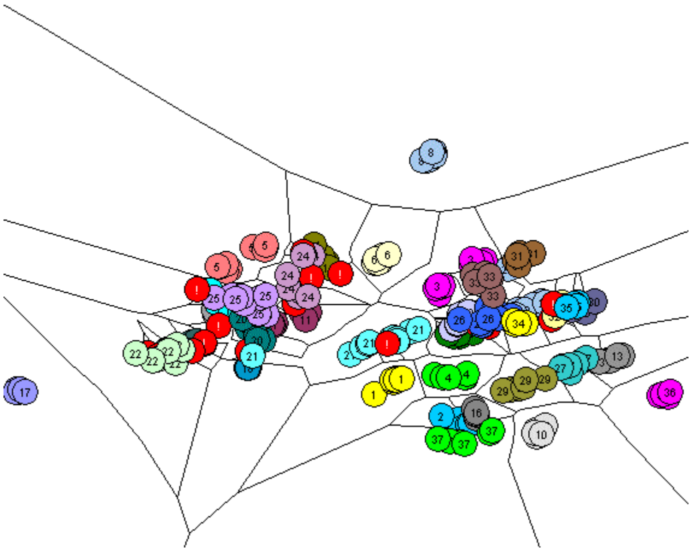
Classification rate: 72.348 %

K-NN (Special Tube holder)

Training



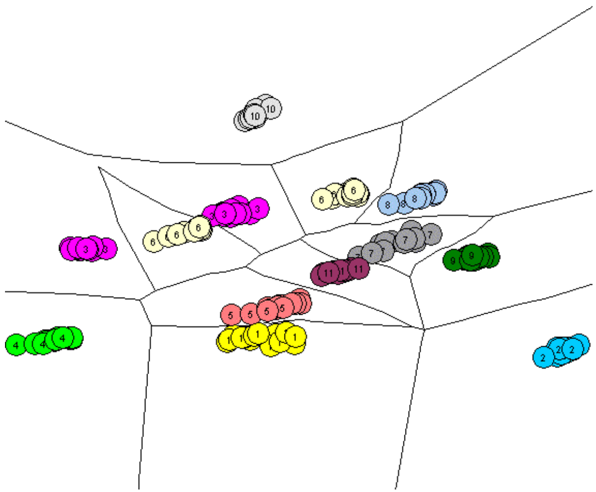
Testing



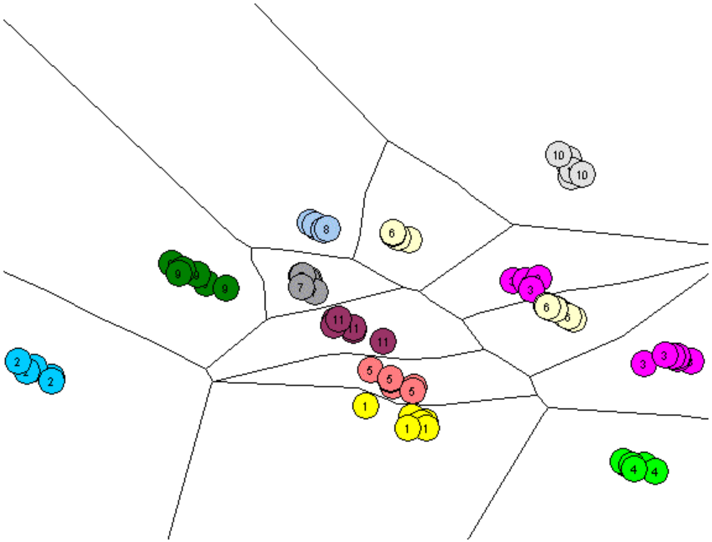
Classification rate: 92.199 %

K-NN (Dortrecht)

Training



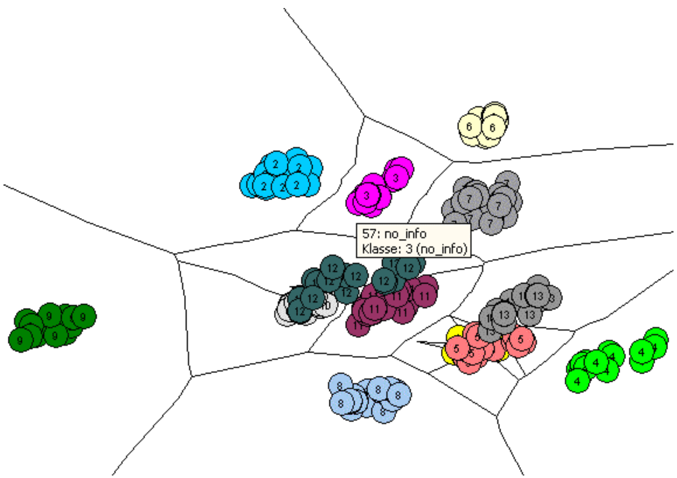
Testing



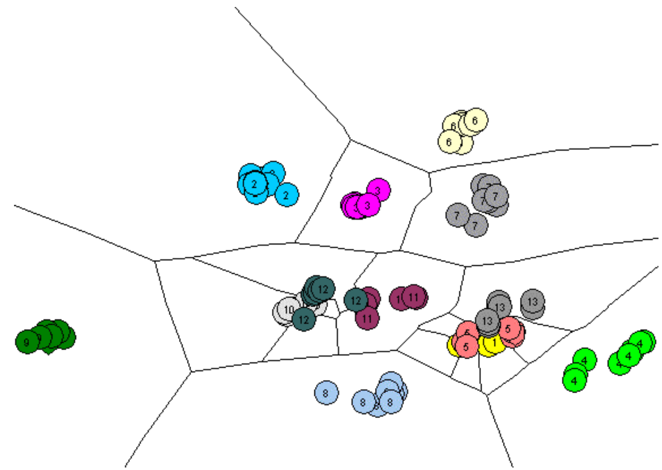
Classification rate: 100.000 %

K-NN (Validierung & Analyse)

Training



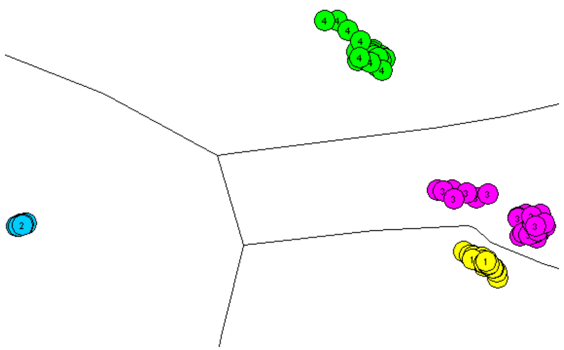
Testing



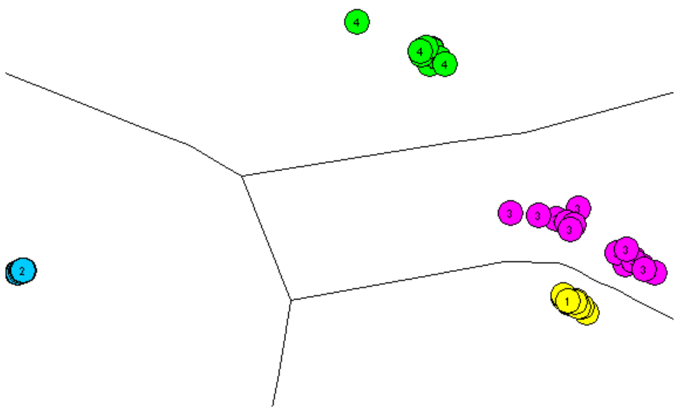
Classification rate: 100.000 %

K-NN (OLA Goldach)

Training



Testing



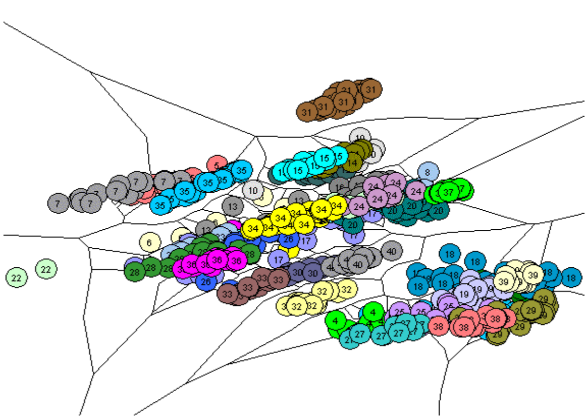
Confusion Matrix,1

Class1 (15):	0 (R)	100	0	0	0
Class2 (15):	0 (R)	0	100	0	0
Class3 (15):	0 (R)	0	0	100	0
Class4 (15)	0 (R)	0	0	0	100

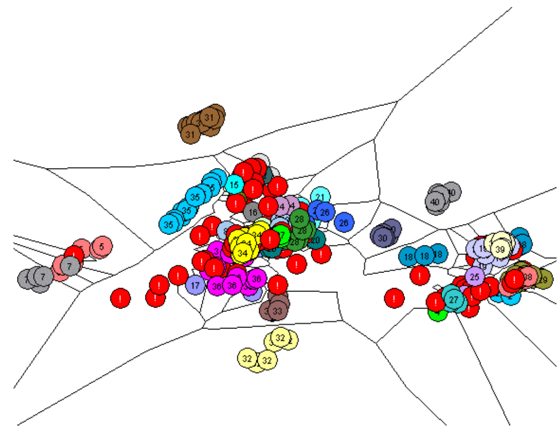
Classification rate: 100.000 %

PNN (Tiger cap set)

Training



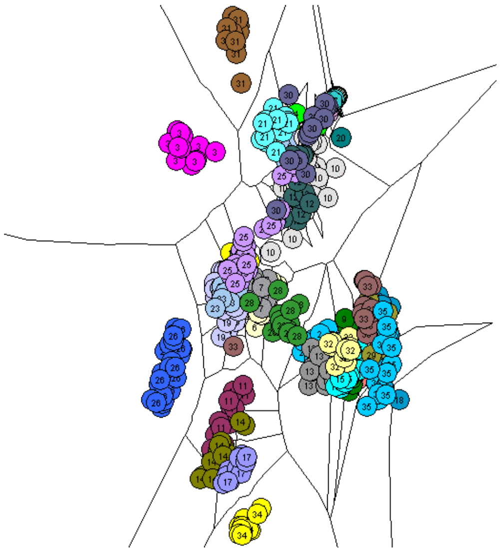
Testing



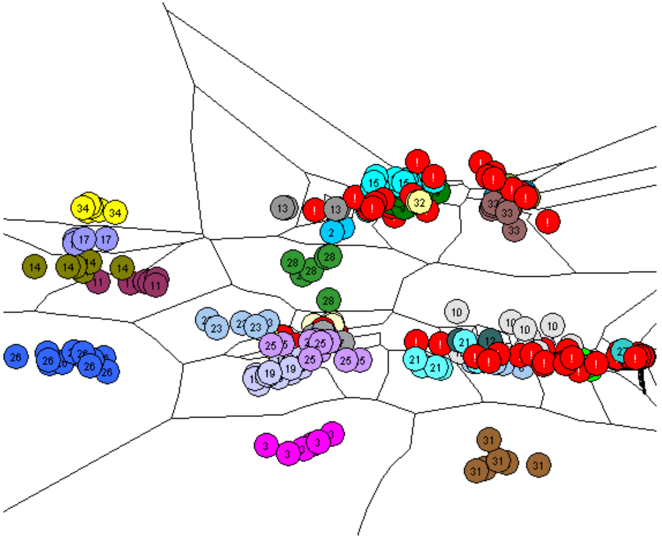
Classification rate: 65.965 %

PNN (Tiger cap set top only)

Training



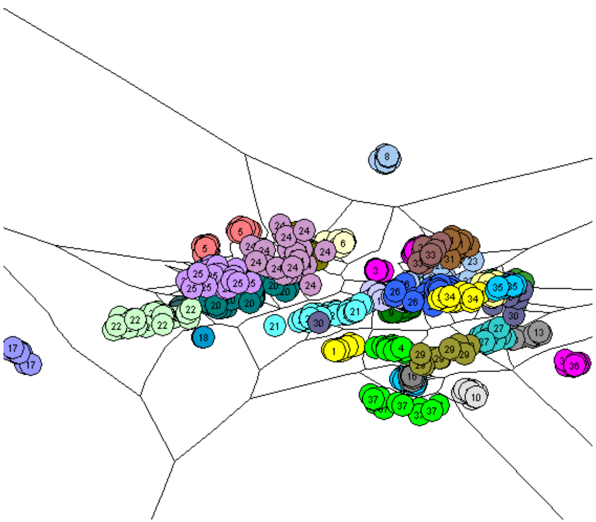
Testing



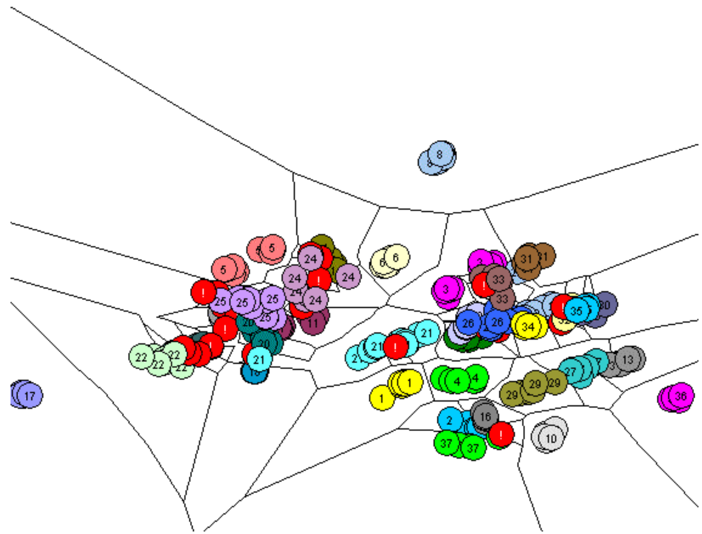
Classification rate: 62.500 %

PNN (Special Tube holder)

Training



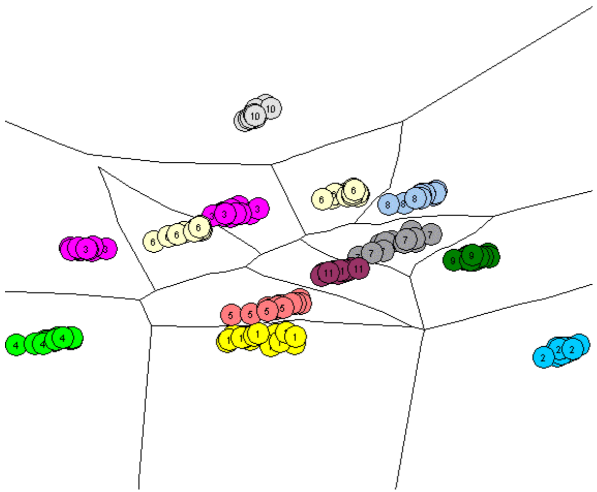
Testing



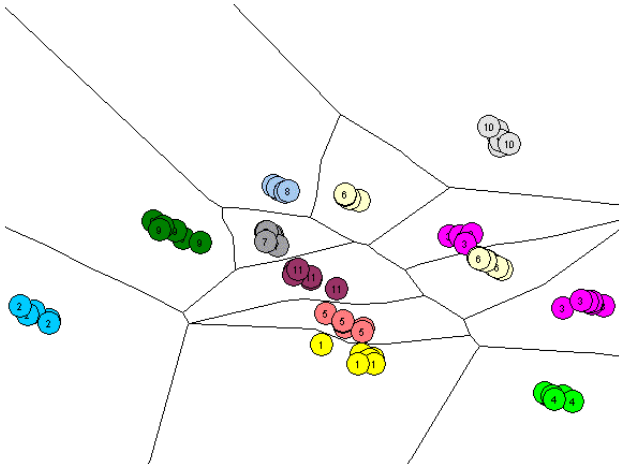
Classification rate: 87.589 %

PNN (Dortrecht)

Training



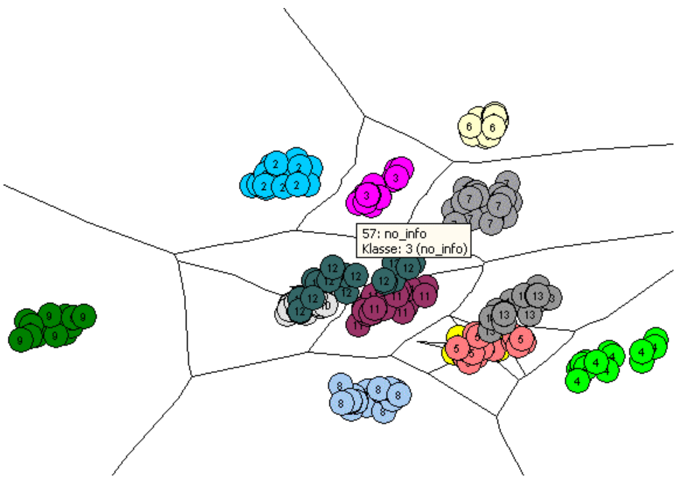
Testing



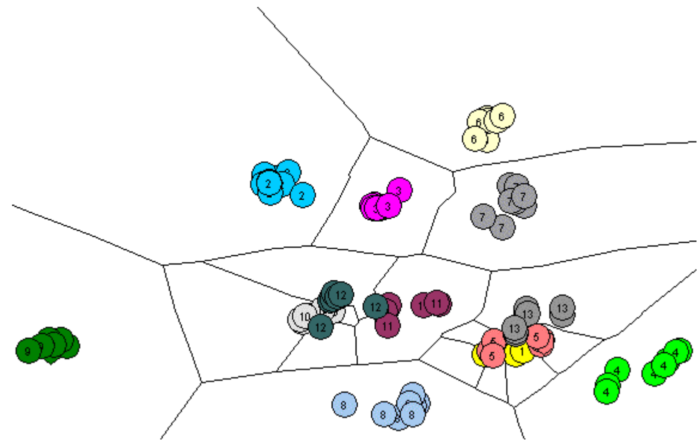
Classification rate: 100.000 %

PNN (Validierung & Analyse)

Training



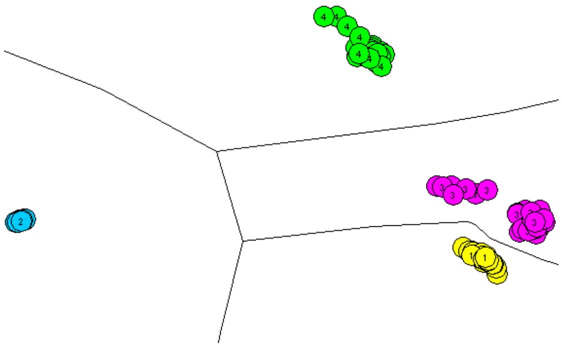
Testing



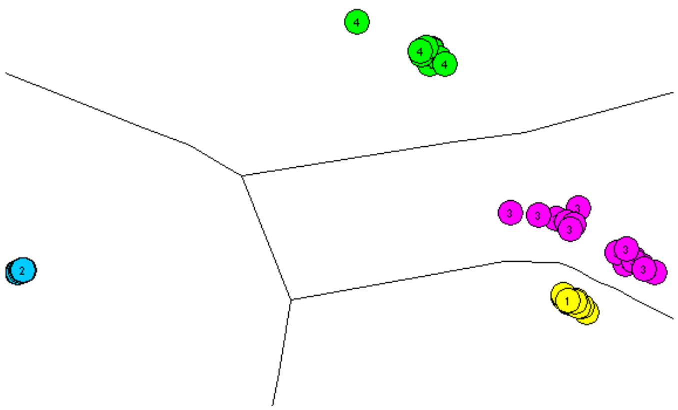
Classification rate: 100.000 %

PNN (OLA Goldach)

Training



Testing



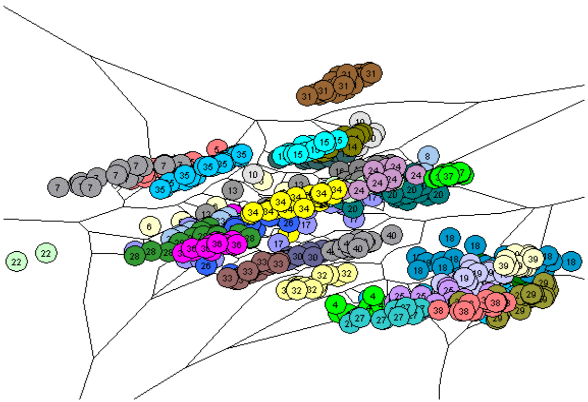
Confusion Matrix,1

Class1 (15):	0 (R)	100	0	0	0
Class2 (15):	0 (R)	0	100	0	0
Class3 (15):	0 (R)	0	0	100	0
Class4 (15)	0 (R)	0	0	0	100

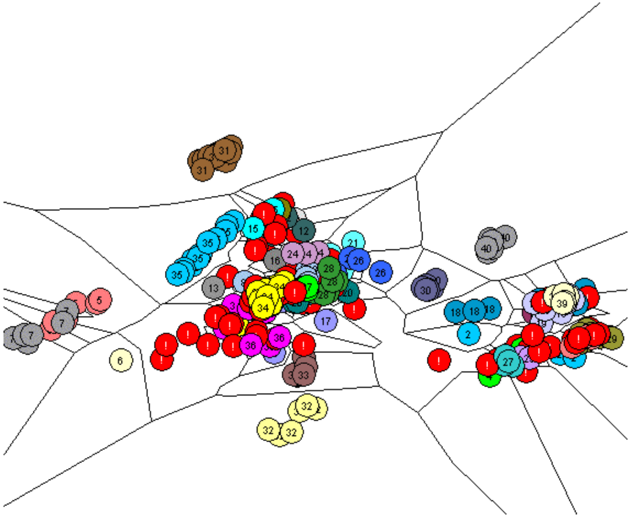
Classification rate: 100.000 %

RNN (Tiger cap set)

Training



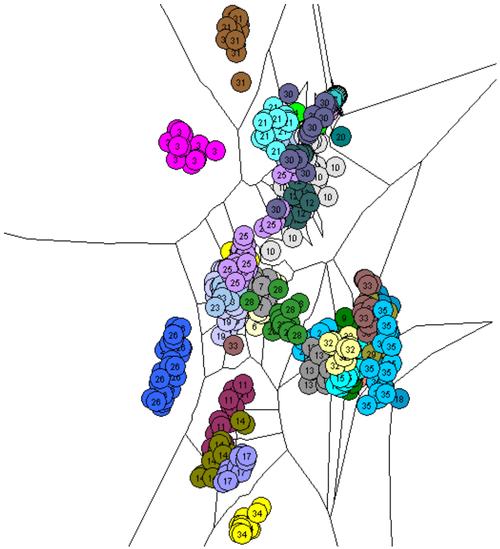
Testing



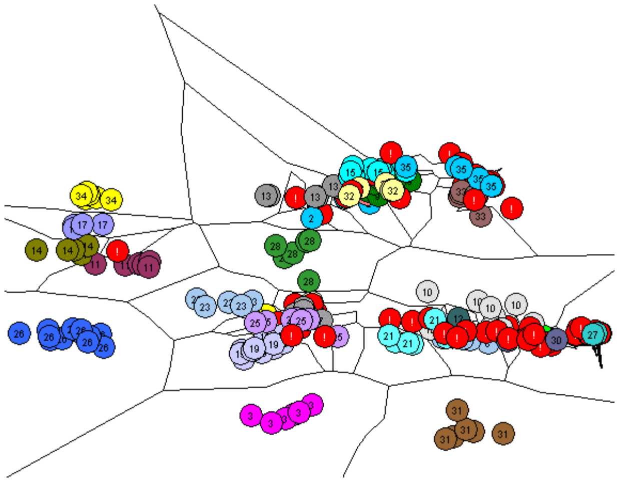
Classification rate: 68.070 %

RNN (Tiger cap set top only)

Training



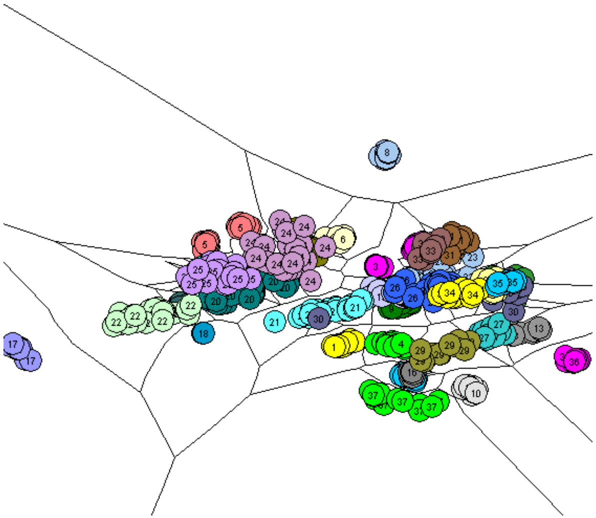
Testing



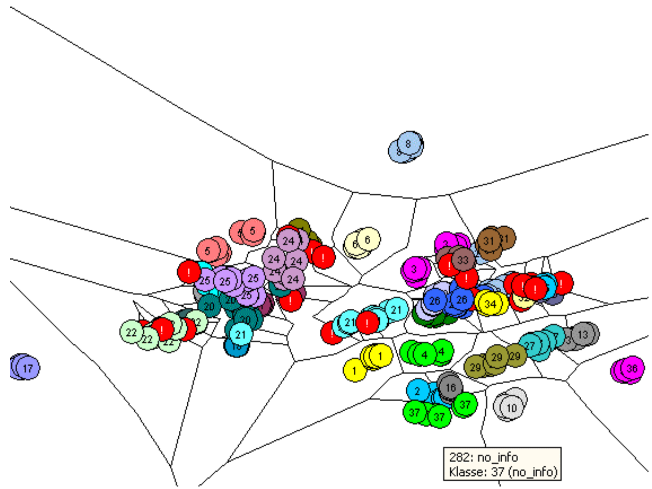
Classification rate: 68.561 %

RNN (Special Tube holder)

Training



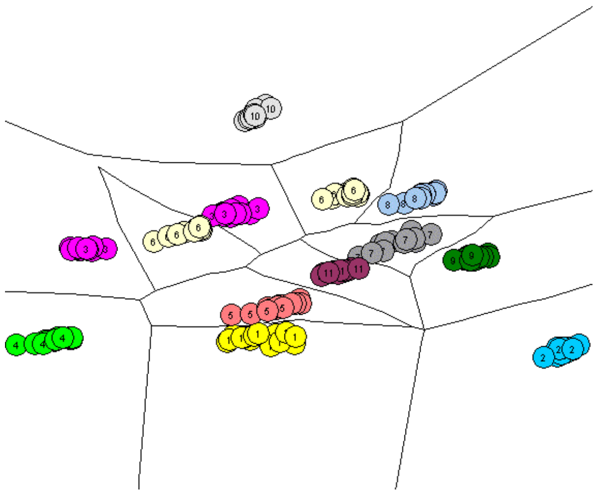
Testing



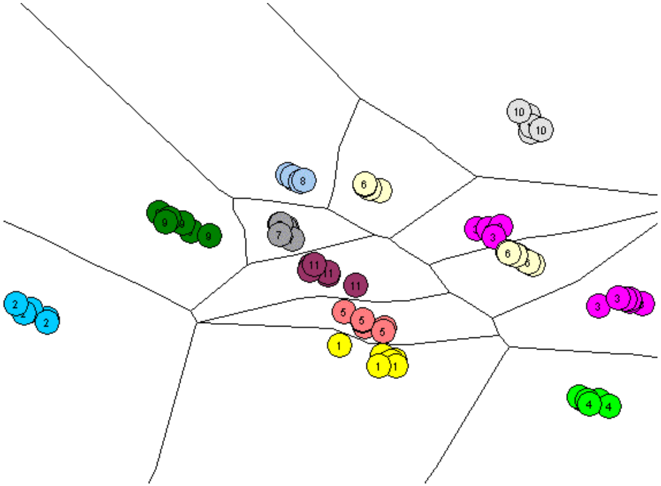
Classification rate: 90.426 %

RNN (Dortrecht)

Training



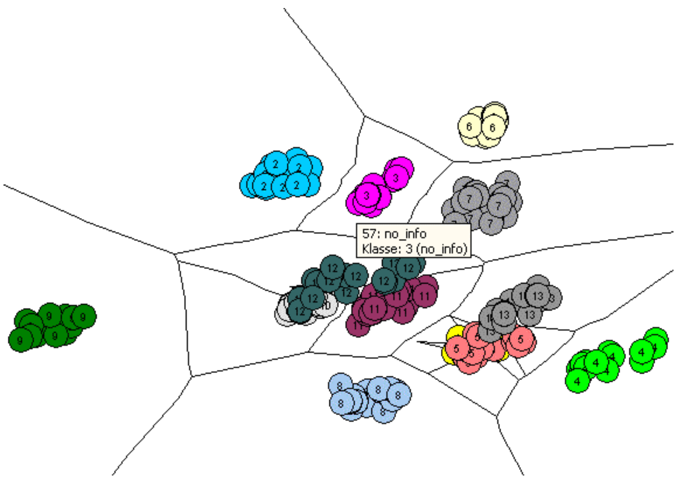
Testing



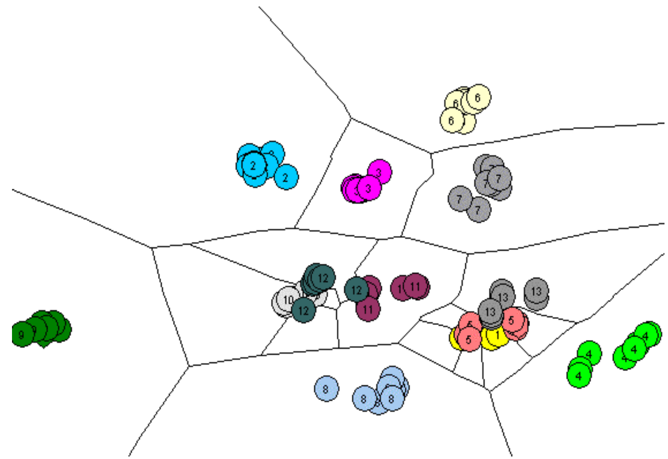
Classification rate: 100.000 %

RNN (Validierung & Analyse)

Training



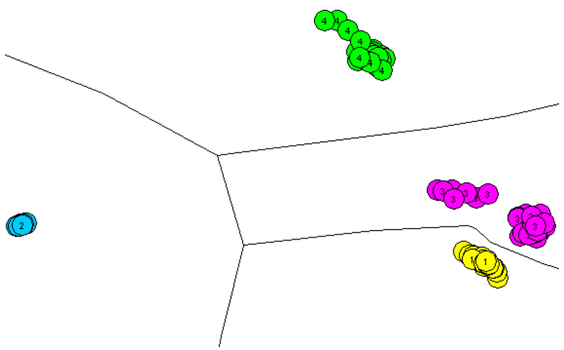
Testing



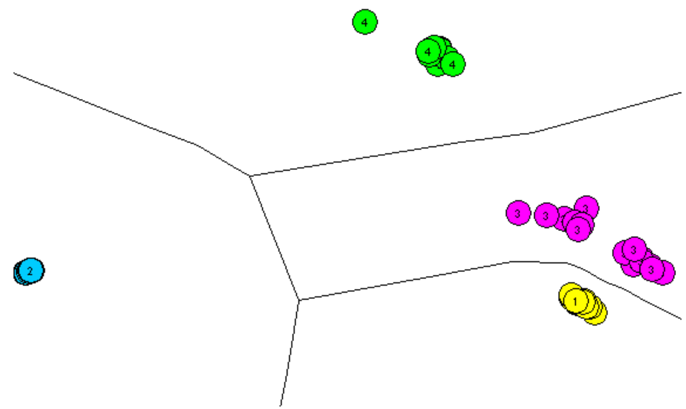
Classification rate: 100.000 %

RNN (OLA Goldach)

Training



Testing



Confusion Matrix,1

Class1 (15):	0 (R)	100	0	0	0
Class2 (15):	0 (R)	0	100	0	0
Class3 (15):	0 (R)	0	0	100	0
Class4 (15)	0 (R)	0	0	0	100

Classification rate: 100.000 %

Classsification on Robot Vision data

Data	Gamma	Sigma	LS-SVM	K-NN (K=5)	PNN	RNN
Tiger-capset	2	0.005	56.14%	67.368%	T=1 ,Sig=0.1 65.965%	68.07%
Tigercap-top only	0.7	0.02	55.303%	72,348%	T=1,Sig=0.1 62.5%	68.56%
SPTH	0.8	0.02	78.723%	92.19%	T=1, Sig=0.1 87.58%	90.426%
Dortrecht	2	0.01	100%	100%	T=1,Sig=0.1 100%	100%
VA	2.0554	0.0062523	98.039%	100%	T=1,Sig=0.1 100%	100%
OLA Goldach	4	2	100%	100%	T=1,Sig=0.1 100%	100%

Classification using SVM

- Unsatisfied result using LSSVM lead us to SVM
- SVM uses Quadratic Programming
- We used the same data for classification
- SVM has different parameters
- We used Manual tuning
- Tuning steps as same as LSSVM

Classification results using SVM

Data	C	Sigma	CR(%)
Tiger cap set	10	0.05	70.175
Tiger cap set top only	15	0.05	74.24
Special tube holder	15	0.05	92.199
Dorthrecht	10	0.05	100
Validierund & Analyse	20	0.03	100
OLA Goldach (One Vs All)	10	0.05	100

LIBSVM Vs Other Classifiers

Data	LS-SVM	K-NN (K=5)	PNN	RNN	SVM
Tiger-capset	56.14%	67.368%	T=1 ,Sig=0.1 65.965%	68.07%	70.175%
Tigercap-top only	55.303%	72,348%	T=1,Sig=0.1 62.5%	68.56%	74.24%
SPTH	78.723%	92.19%	T=1, Sig=0.1 87.58%	90.426%	92.199%
Dortrecht	100%	100%	T=1,Sig=0.1 100%	100%	100%
VA	98.039%	100%	T=1,Sig=0.1 100%	100%	100%
OLA Goldach	100%	100%	T=1,Sig=0.1 100%	100%	100%

Conclusion

- Even though LS-SVM is a powerful tool for classification and regression, in our experiment, only with Repository data LS-SVM seems to be competitive with other classifiers
- For the Robot vision data, for Dortrecht data set, VA and OLA, LS-SVM is comparable with other classifiers
- For rest of the data sets LS-SVM is seems to be not well generalizing
- As the complexity of the data set increases, generalizing capability of LS-SVM has to be checked
- SVM with QP proved to be the best for our multi class data

References

- Lecture documents of *Sensorsignalverarbeitung* from Prof.Dr.-Ing Andreas König
- Quickcog user manual
- Hold out algorithm from Mr. Kuncup Iswandy
- LS-SVM MATLAB toolbox and user manual
- SVM toolbox (Source Mr.Iswandy)
- Support Vector Machines Andrew W. Moore Professor (School of Computer Science Carnegie Mellon University)

<u>THANK YOU FOR YOR ATTENTION</u>

????????????????