A comparison of machine learning algorithms applied to hand gesture recognition

Paulo Trigueiros
Departamento de Informática
Instituto Politécnico do Porto
Porto, Portugal
paulo@trigueiros.org

Fernando Ribeiro
Departamento de Electrónica Industrial
Campus de Azurém 4800-058
Universidade do Minho
Guimarães, Portugal
fernando@dei.uminho.pt

Luís Paulo Reis [1]
[1] EEUM – Escola de Engenharia da Universidade do Minho - DSI
Campus de Azurém 4800-058
Guimarães, Portugal
lpreis@dsi.uminho.pt

Abstract—**Hand gesture recognition for human computer interaction is an area of active research in computer vision and machine learning. The primary goal of gesture recognition research is to create a system, which can identify specific human gestures and use them to convey information or for device control. This paper presents a comparative study of four classification algorithms for static hand gesture classification using two different hand features data sets. The approach used consists in identifying hand pixels in each frame, extract features and use those features to recognize a specific hand pose. The results obtained proved that the ANN had a very good performance and that the feature selection and data preparation is an important phase in the all process, when using low-resolution images like the ones obtained with the camera in the current work.**

Keywords - Machine vision, image processing, machine learning, hand gesture recognition.

I. Introduction

Gesture recognition is an area of active current research in computer vision and machine learning [1].

Hand gesture recognition, being a natural way of human computer interaction, is an area where many researchers in the academia and industry are working on different applications to make interactions more easy, natural and convenient without wearing any extra devices [2].

Human-computer interfaces (HCI) have evolved from text-based interfaces through 2D graphical-based interfaces, multimedia-supported interfaces, to fully-fledged multi-participant Virtual Environment (VE) systems. While providing a new sophisticated paradigm for human communication, interaction, learning and training, VE systems also provide new challenges since they include many new types of representation and interaction [3].

To achieve natural human-computer interaction for VE applications, the human hand could be considered as an input device. Hand gestures are a powerful human communication modality with lots of potential applications. However, the expressiveness of hand gestures has not been fully explored for HCI applications. Compared with traditional HCI devices, hand gestures are less intrusive and more convenient to explore the three-dimensional (3D) virtual worlds.

The approach used for the problem in hands consists of identifying the pixels on the image that constitute the hand, extract features from those identified pixels in order to classify the hand, and use those features to recognize the occurrence of specific pose sequences as gestures.

Several algorithms exist that can be applied to the task of learning from hand features.

This paper presents a comparative study of four learning algorithms applied to two different hand datasets: k-Nearest Neighbour (k-NN), Naïve Bayes (NB), Artificial Neural Network (ANN) and Support Vector Machines (SVM). We present and explain the steps taken to extract hand features from a series of consecutive frames, in order to classify the data to obtain results and conclusions. This study allowed the implementation of some algorithms and the application to one simple classification problem: static hand gesture recognition. The tool used for developing the machine learning and data mining experiments was Rapid Miner [4].

The rest of the paper is as follows. First we review related work in section 2. Section 3 introduces machine learning. The actual data pre-processing stage and feature extraction is described in section 4. Dataset structure and experimental methodology is explained in section 5. Section 6 presents the results obtained with the two datasets and discusses the values obtained. Conclusions and future work are draw in section 7.

II. Related Work

There are many studies on gesture recognition and methodologies well presented in [5, 6]. Machine learning algorithms have been applied successfully to many fields of research like, face recognition [7], automatic recognition of a musical gesture by a computer [8], classification of robotic soccer formations [9], classifying human physical activity from on-body accelerometers [10] and automatic road-sign detection [11, 12].

K-Nearest Neighbour was used in [7, 9]. This classifier represents each example as a data in d–dimensional space, where d is the number of attributes. Given a test example it is computed the proximity to the rest data points in the training set using a measure of similarity or dissimilarity. In the distance calculation, the standard Euclidean distance is normally used, however other metrics can be used [13]. An artificial neural network is a mathematical/computational model that attempts to simulate the structure of biological neural systems. They accept features as inputs and produce decisions as outputs [14]. Maung et al [1, 9, 12, 15] used it in a gesture recognition system, Faria et al [9] used it for the classification of robotic soccer formations, Vicen-Buéno [12] used it applied to the problem of traffic sign recognition and Stephan et al used it for static hand gesture recognition for human-computer interaction. Support Vector Machines (SVM's) is a technique based on statistical learning theory, which works very well with high-dimensional data. The objective of this algorithm is to find the optimal separating hyper plane between two classes by maximizing the margin between them [16]. Faria et al. [7, 9] used it to classify robotic soccer formations and the classification of facial expressions, Ke et al. [17] used it in the implementation of a real-time hand gesture recognition system for human robot interaction, Maldonado-Báscon [11] used it for the recognition of road-signs and Masaki et al used it in conjunction with SOM (Self-Organizing Map) for the automatic learning of a gesture recognition mode

III. MACHINE LEARNING

The study and computer modelling of learning processes in their multiple manifestations constitutes the topic of machine learning [18].

Machine learning is the task of programming computers to optimize a performance criterion using example data or past experience [19]. For that, machine learning uses statistic theory in building mathematical models, because the core task is to make inference from sample data.

In machine learning two entities, the teacher and the learner, play a crucial role. The teacher is the entity that has the required knowledge to perform a given task. The learner is the entity that has to learn the knowledge to perform the task. We can distinguish learning strategies by the amount of inference the learner performs on the information provided by the teacher.

The learning problem can be stated as follows: given an example set of limited size, find a concise data description [18]. Given this, learning techniques can be grouped in three big families: supervised learning, reinforcement learning and unsupervised learning.

In supervised learning, given a sample of input-output pairs, called the training sample, the task is to find a deterministic function that maps any input to an output that can predict future observations, minimizing the error as much as possible. According to the type of outputs, supervised learning can be distinguished in classification and regression learning [19]. In classification problems the task is to assign new inputs to one of a number of discrete classes or categories. If the output space is formed by the values of a continuous variable, then the learning task is known as the problem of regression or function learning [20].

In reinforcement learning the problem is to learn what to do, i.e. how to map situations to actions, in order to maximize a given reward. The learning algorithm is not told which actions to take in a given situation, instead, the learner is assumed to gain information about the actions taken by some reward, not necessarily arriving immediately after the action is taken. In order to maximize reward a learning algorithm must choose actions, which have been tried out in the past and found to be effective in producing reward (exploit current knowledge). On the other hand, to discover those actions the learning algorithm has to choose actions not tried in the past (explore the state space).

In unsupervised learning, the data is only a sample of objects without associated target values. Here, a concise description of the data can be a set of clusters or a probability density stating how likely it is to observe a certain object in the future [21]. The goal in unsupervised learning is, given a training sample of objects extract some structure from them.

In our study, the classes for classification are known in advance and the objective is to classify each new observation to the respective class using a specific function.

IV. PRE-PROCESSING AND FEATURE EXTRACTION

Hand segmentation and feature extraction is a crucial step in computer vision applications for hand gesture recognition. The pre-processing stage prepares the input image and extracts features used later with the classification algorithms.

In the present study, we used two data sets with different features extracted from the segmented hand. The hand features used for the training datasets are: the mean and variance of the grey pixel values (Figure 1), the blob area, perimeter and the number of convexity defects (Figure 2), the hand orientation, the orientation histogram (Figure 3) and the radial signature (Figure 4)

Figure 1. Detected hand with mean and variance grey values

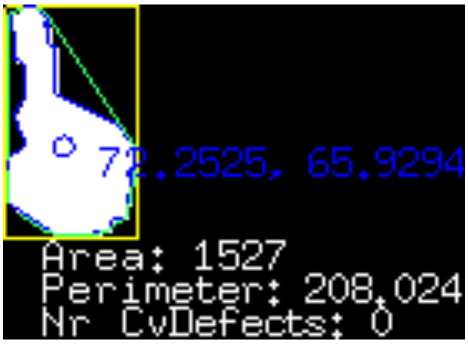

Figure 2. Hand blob

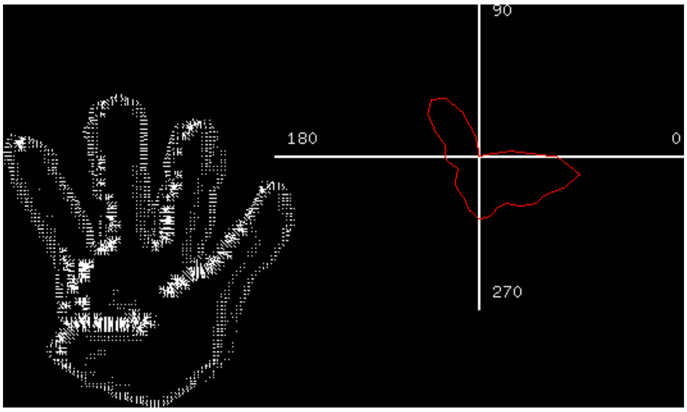

Figure 3. Hand gradients and orientation histogram

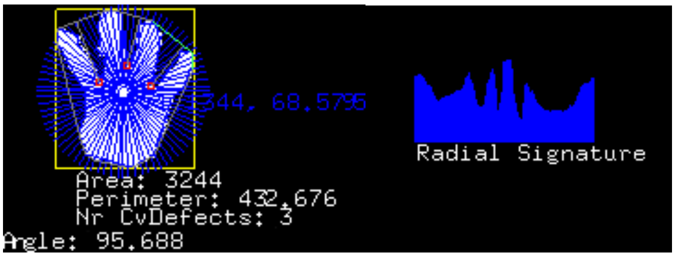

Figure 4. Hand radials and the respective radial signature

A. Hand orientation

The segmented hand is used to calculate its orientation with the help of image moments. Moment approximations provide one of the more obvious means of describing 2-D shapes [22]. The moments involve sums over all pixels, and so are robust against small pixel changes [23]. If $I(x,y)$ is the image intensity at position x and y then the image moments, up to the second order, are:

$$M_{00} = \sum_x \sum_y I(x,y) \quad M_{11} = \sum_x \sum_y xy.I(x,y)$$

$$M_{10} = \sum_x \sum_y x.I(x,y) \; M_{01} = \sum_x \sum_y y.I(x,y) \tag{1}$$

$$M_{20} = \sum_x \sum_y x^2.I(x,y) \; M_{02} = \sum_x \sum_y y^2.I(x,y)$$

The hand position can be calculated as follows:

$$x_c = \frac{M_{10}}{M_{00}} \; y_c = \frac{M_{01}}{M_{00}} \tag{2}$$

The hand orientation can then be calculated using the following intermediate variables a, b and c:

$$a = \frac{M_{20}}{M_{00}} - x_c^2$$

$$b = 2\left(\frac{M_{10}}{M_{00}} - x_c y_c\right) \tag{3}$$

$$c = \frac{M_{02}}{M_{00}} - y_c^2$$

and the angle is:

$$\theta = \frac{\arctan(b,(a-c))}{2} \tag{4}$$

B. Orientation histogram

Pixel intensities can be sensitive to lighting variations, which lead to classification problems within the same gesture under different light conditions. The use of local orientation measures avoids this kind of problem, and the histogram gives us translation invariance. Orientation histograms summarize how much of each shape is oriented in each possible direction, independent of the position of the hand inside the camera frame [23].

This statistical technique is most appropriate for close-ups of the hand. In our work, the hand is extracted and separated from the background, which provides a uniform black background, which makes this statistical technique a good method for the identification of different static hand poses.

This method is insensitive to small changes in the size of the hand, but it is sensitive to changes in hand orientation.

We have calculated the local orientation using image gradients, represented by horizontal and vertical image pixel differences. If d_x and d_y are the outputs of the derivative operators, then the gradient direction is arctan (d_x, d_y), and the contrast is $\sqrt{d_x^2 + d_y^2}$. A contrast threshold is set as some amount k times the mean image contrast, below which we assume the orientation measurement is inaccurate. A value of k=1.2 was used in the experiments. We then blur the histogram in the angular domain as in [24], with a [1 4 6 4 1] filter, which gives a gradual fall-off in the distance between orientation histograms.

C. Radial signature

A simple method to assess the gesture would be to measure the distance from the hand centroid to the edges of the hand along a number of equally spaced radials [25]. For the present feature extraction problem, 100 equally spaced radials were used. To count the number of pixels along a given radial we only take into account the ones that are part of the hand, eliminating those that fall inside gaps, like the ones that appear between fingers or between the palm and a finger. All the radial measurements can be scaled so that the longest radial has a constant length. With this measure, we can have a radial length signature that is invariant to hand distance from the camera.

V. DATASET AND EXPERIMENTAL METHODOLOGY

For machine learning applications, feature selection, data set preparation and data transformation is an important phase. To construct the right model it is necessary to understand the data. Successful data mining involves far more than selecting a learning algorithm and running it over your data [13].

Two different datasets with a different set of hand features have been used to test the machine learning algorithms.

The first dataset has the hand angle, the mean and variance of the segmented hand grey image, the area and perimeter of the binary hand blob and the number of convexity defects. The second dataset has the hand angle, the mean and variance of the segmented hand grey image, the 36 bin values of the orientation histogram, and the 100 bin values of the hand radial signature.

The first step for organizing the data was the creation of a text file with the hand features. For that it was created an application in C++ that is connected to a kinect[1] camera, and the grey image values and the depth image values are used for feature extraction.

After that, the text file is converted to a .xls file to be imported with the Rapid Miner application for performance testing with the four selected algorithms.

A. Experiments

The experimental results were achieved using RapidMiner 5.2 [4] in an Intel Core i7 (2,8 GHz) Mac OSX computer with 4GB DDR3. The processing times were measured and compared in the different experiences.

The four algorithms (k-NN, Naïve Bayes, ANN, SVM) were applied to the two datasets, and the experiments were performed under the assumption of the k-fold method. The k-fold cross validation is used to determine how accurately a learning algorithm will be able to predict data that it was not trained with [7]. In the k-fold cross-validation, the dataset X is divided randomly into k equal sized parts, X_i, $i = 1, \dots, k$. The learning algorithm is then trained k times, using k-1 parts as the training set and the one that stays out as the validation set. A value of k=10 is normally used, giving a good rule of approximation, although the best value depends on the used algorithm and the dataset [13, 19].

Prior to the application of the algorithms the data was standardized by the application of the Z score,

$$Z = \frac{(a_{ij} - \bar{a})}{\sigma} \quad (5)$$

where $\bar{a}$ is the mean of attribute a_i and σ is the respective standard deviation.

Then, the performance of the algorithms, based on the number of counts of test records correctly and incorrectly predicted by the model, was analysed.

The settings and parameters used with the classifiers is an important aspect to take into account. Thus, for the simplest of the algorithms, the k-NN a parameter optimization analysis was made, given a value of k=1 for the two datasets, and a Euclidean distance numerical measure.

The neural network operator learns a model by means of a feed-forward neural network trained by a back propagation algorithm (multi-layer perceptron). For the experiments the following parameters (Table 1) were used:

TABLE 1 PARAMETERS USED WITH THE NEURAL NETWORK

Parameters	Values	
	DataSet 1	DataSet 2
Learning rate	0,1	0,33
Momentum rate	0,1	0,18
Training cycles	500	500
Hidden layers in the network	1	1

For the tests with SVM the libSVM[2] algorithm was used that is incorporated in the RapidMiner, which supports multiclass learning. For experimental results the type of SVM used was C-SCV, and the kernel was sigmoid for dataset 2 and RBF for dataset 1 has can be seen on Table 2.

TABLE 2 PARAMETERS USED WITH THE SVM (LIBSVM)

Parameters	Values	
	DataSet 1	Data Set 2
Kernel type	Rbf	sigmoid
Cost parameter for C-SVC	10	0

The parameters presented in both tables were obtained with Rapid Miner using a 10-fold cross validation on the two datasets with parameter optimization for the NN and SVM algorithms.

[1] http://pt.wikipedia.org/wiki/Kinect

[2] http://www.csie.ntu.edu.tw/~cjlin/libsvm/

VI. RESULTS AND DISCUSSION

The results obtained with the two datasets are represented in Table 3.

TABLE 3 ACCURACY AND TIME OF EXPERIENCE (DATASET 1 AND 2)

	Classifier	k-NN	Naïve Bayes	ANN	SVM
DataSet 1	Accuracy (%)	95,45	25,87	96,99	91,66
	Time	8''	1''	46'33''	3'10''
DataSet 2	Accuracy (%)	88,52	66,50	85,18	80,02
	Time	1''	1''	32''	1'08''

In terms of accuracy the learning process using only five features has lower accuracy than the dataset with more features. It can be observed by the above table that the time that takes to train the neural network is the highest one but has the best results.

To analyze how classification errors are distributed among classes we computed for each trained algorithm it's corresponding confusing matrix has shown in the following figures.

	true 1	true 2	true 3	true 4	true 5	true 6	true 7	true 8	true 9	true 10	class precisic
pred. 1	91	1	0	0	1	0	0	0	0	1	96.81%
pred. 2	0	131	0	0	0	1	1	0	0	5	94.93%
pred. 3	1	0	122	1	0	0	0	2	0	1	96.06%
pred. 4	0	0	0	102	3	1	0	0	0	0	96.23%
pred. 5	0	0	1	2	123	0	0	0	0	0	97.62%
pred. 6	0	0	0	0	0	122	7	0	0	0	94.57%
pred. 7	0	0	0	0	0	5	120	0	0	0	96.00%
pred. 8	0	0	1	0	0	0	0	146	0	0	99.32%
pred. 9	1	0	0	0	0	0	0	0	155	0	99.36%
pred. 10	0	2	0	1	0	0	0	0	0	144	97.96%
class recall	97.85%	97.76%	98.39%	96.23%	96.85%	94.57%	93.75%	98.65%	100.00%	95.36%	

Figure 5 k-NN DataSet 1

	true 1	true 2	true 3	true 4	true 5	true 6	true 7	true 8	true 9	true 10	class precisio
pred. 1	59	1	2	0	0	0	0	0	4	1	88.06%
pred. 2	0	74	0	0	0	0	0	0	0	1	98.67%
pred. 3	2	0	49	1	2	0	0	3	1	3	80.33%
pred. 4	0	0	1	38	10	0	0	1	0	2	73.08%
pred. 5	0	0	0	6	52	6	2	0	0	2	76.47%
pred. 6	0	0	0	0	5	68	1	0	1	1	89.47%
pred. 7	0	0	0	0	0	1	86	0	0	0	98.85%
pred. 8	7	0	5	1	1	0	0	92	0	0	86.79%
pred. 9	0	0	0	0	0	1	0	0	82	0	98.80%
pred. 10	2	1	5	0	2	1	0	2	1	87	86.14%
class recall	84.29%	97.37%	79.03%	82.61%	72.22%	88.31%	96.63%	93.88%	92.13%	89.69%	

Figure 6. k-NN DataSet 2

	true 1	true 2	true 3	true 4	true 5	true 6	true 7	true 8	true 9	true 10	class precisio
pred. 1	11	4	6	0	0	1	7	7	0	1	29.73%
pred. 2	14	49	0	24	25	24	20	0	12	24	25.52%
pred. 3	25	4	24	27	66	55	54	7	65	21	6.90%
pred. 4	0	0	0	0	0	0	0	0	0	0	0.00%
pred. 5	0	10	0	0	1	14	9	0	0	2	2.78%
pred. 6	0	3	0	0	0	0	0	0	0	0	0.00%
pred. 7	6	6	18	4	4	6	10	8	16	0	12.82%
pred. 8	33	0	73	47	24	15	5	123	43	5	33.42%
pred. 9	3	0	3	0	0	0	1	3	19	0	65.52%
pred. 10	1	58	0	4	7	14	22	0	0	98	48.04%
class recall	11.83%	36.57%	19.35%	0.00%	0.79%	0.00%	7.81%	83.11%	12.26%	64.90%	

Figure 7. Naive Bayes DataSet 1

	true 1	true 2	true 3	true 4	true 5	true 6	true 7	true 8	true 9	true 10	class precisio
pred. 1	62	0	0	0	0	0	0	3	11	1	80.52%
pred. 2	2	74	0	0	0	0	0	0	0	2	94.87%
pred. 3	2	0	54	1	0	0	0	7	2	5	76.06%
pred. 4	0	0	2	34	5	0	0	0	0	1	80.95%
pred. 5	0	0	1	8	64	8	2	0	2	2	73.56%
pred. 6	0	0	0	1	1	67	3	0	1	4	87.01%
pred. 7	0	0	0	0	1	2	84	0	0	1	95.45%
pred. 8	4	1	3	2	0	0	0	79	0	9	80.61%
pred. 9	0	0	0	0	0	0	0	0	72	1	98.63%
pred. 10	0	1	2	0	1	0	0	9	1	71	83.53%
class recall	88.57%	97.37%	87.10%	73.91%	88.89%	87.01%	94.38%	80.61%	80.90%	73.20%	

Figure 8. Naive Bayes DataSet 2

	true 1	true 2	true 3	true 4	true 5	true 6	true 7	true 8	true 9	true 10	class precisio
pred. 1	91	0	4	0	1	0	0	0	2	0	92.86%
pred. 2	0	129	0	0	1	1	2	0	0	1	96.27%
pred. 3	0	1	117	0	0	0	1	0	0	2	96.69%
pred. 4	0	0	3	103	1	0	0	0	0	0	96.26%
pred. 5	1	2	0	2	122	2	2	0	0	1	92.42%
pred. 6	0	0	0	0	1	125	0	0	0	0	99.21%
pred. 7	0	0	0	1	0	1	122	0	1	0	97.60%
pred. 8	1	0	0	0	0	0	0	148	0	0	99.33%
pred. 9	0	0	0	0	0	0	1	0	152	0	99.35%
pred. 10	0	2	0	0	1	0	0	0	0	147	98.00%
class recall	97.85%	96.27%	94.35%	97.17%	96.06%	96.90%	95.31%	100.00%	98.06%	97.35%	

Figure 9. NN DataSet 1

	true 1	true 2	true 3	true 4	true 5	true 6	true 7	true 8	true 9	true 10	class precisio
pred. 1	62	0	0	0	0	0	0	3	11	1	80.52%
pred. 2	2	74	0	0	0	0	0	0	0	2	94.87%
pred. 3	2	0	54	1	0	0	0	7	2	5	76.06%
pred. 4	0	0	2	34	5	0	0	0	0	1	80.95%
pred. 5	0	0	1	8	64	8	2	0	2	2	73.56%
pred. 6	0	0	0	1	1	67	3	0	1	4	87.01%
pred. 7	0	0	0	0	1	2	84	0	0	1	95.45%
pred. 8	4	1	3	2	0	0	0	79	0	9	80.61%
pred. 9	0	0	0	0	0	0	0	0	72	1	98.63%
pred. 10	0	1	2	0	1	0	0	9	1	71	83.53%
class recall	88.57%	97.37%	87.10%	73.91%	88.89%	87.01%	94.38%	80.61%	80.90%	73.20%	

Figure 10. NN DataSet 2

	true 1	true 2	true 3	true 4	true 5	true 6	true 7	true 8	true 9	true 10	class precisio
pred. 1	85	0	0	0	0	4	0	1	0	1	93.41%
pred. 2	0	116	0	4	0	1	1	0	0	3	92.80%
pred. 3	0	0	115	7	1	1	0	0	0	1	92.00%
pred. 4	0	0	3	86	11	3	1	0	0	1	81.90%
pred. 5	3	0	0	9	112	10	2	0	0	1	81.75%
pred. 6	0	2	0	0	0	109	4	0	0	0	94.78%
pred. 7	0	0	0	0	0	1	119	0	0	0	99.17%
pred. 8	4	0	6	0	0	0	0	146	0	0	93.59%
pred. 9	1	0	0	0	0	0	0	1	155	0	98.73%
pred. 10	0	16	0	0	3	0	1	0	0	144	87.80%
class recall	91.40%	86.57%	92.74%	81.13%	88.19%	84.50%	92.97%	98.65%	100.00%	95.36%	

Figure 11. SVM DataSet 1

	true 1	true 2	true 3	true 4	true 5	true 6	true 7	true 8	true 9	true 10	class precisio
pred. 1	59	0	0	0	0	0	0	0	10	0	85.51%
pred. 2	0	65	0	0	0	0	0	0	0	12	84.42%
pred. 3	3	0	37	0	1	1	0	6	7	2	64.91%
pred. 4	0	0	2	33	2	0	0	0	0	1	86.84%
pred. 5	0	0	0	10	67	9	2	0	0	5	72.04%
pred. 6	0	0	0	0	0	64	2	0	0	1	95.52%
pred. 7	0	0	0	0	0	3	85	0	0	0	96.59%
pred. 8	3	0	13	0	0	0	0	74	0	11	73.27%
pred. 9	0	0	0	0	1	0	0	1	72	0	97.30%
pred. 10	5	11	10	3	1	0	0	17	0	65	58.04%
class recall	84.29%	85.53%	59.68%	71.74%	93.06%	83.12%	95.51%	75.51%	80.90%	67.01%	

Figure 12. SVM DataSet 2

VII. CONCLUSIONS AND FUTURE WORK

Hand gesture recognition is a difficult problem and the current work is only a small step towards achieving the results needed in the field. This paper presented a comparative study of four algorithms applied to two different datasets for static gesture recognition and classification, for human computer interaction. The results achieved identifying hand gestures in the two datasets, enable to conclude that the dataset 1 attributes result better in hand gesture identification, and that the ANN had a very good performance for the problem at hand. The results obtained also proved that the feature selection and data preparation phase is an important one, especially with low-resolution images, which is the case of depth images captured with the kinect camera.

Future work will be concerned with the study of different hand feature selection applied to hand gesture recognition, noise reduction in the depth images acquired, and the introduction of machine learning algorithms based on a combination of methods. An analysis using PCA (principal component analysis) will also be tested, in order to choose the best and simple attributes for classification without losing accuracy.

ACKNOWLEDGMENT

The authors wish to thank all members of the Laboratório de Automação e Robótica, at University of Minho, Guimarães. Also special thanks to the ALGORITMI Research Centre for the opportunity to develop this research work.

REFERENCES

[1] Hninn, T. and H. Maung, Real-Time Hand Tracking and Gesture Recognition System Using Neural Networks. 2009. 50(Frebuary): p. 466-470.

[2] Chaudhary, A., et al., Intelligent Approaches to interact with Machines using Hand Gesture Recognition in Natural way: A Survey. International Journal of Computer Science & Engineering Survey, 2011. 2(1): p. 122-133.

[3] Chen, Q., N.D. Georganas, and E.M. Petriu, Real-time Vision-based Hand Gesture Recognition Using Haar-like Features, in Instrumentation

and Measurement Technology Conference, IEEE, Editor 2007: Warsaw, Poland.

[4] Miner, R. RapidMiner : Report the Future. December 2011]; Available from: http://rapid-i.com/content/view/181/196/.

[5] Mitra, S. and T. Acharya, Gesture recognition: A Survey, in IEEE Transactions on Systems, Man and Cybernetics2007, IEEE. p. 311-324.

[6] Murthy, G.R.S. and R.S. Jadon, A Review of Vision Based Hand Gestures Recognition. International Journal of Information Technology and Knowledge Management, 2009. 2(2): p. 405-410.

[7] Faria, B.M., N. Lau, and L.P. Reis. Classification of Facial Expressions Using Data Mining and machine Learning Algorithms. in 4ª Conferência Ibérica de Sistemas e Tecnologias de Informação. 2009. Póvoa de Varim, Portugal.

[8] Gillian, N.E., Gesture Recognition for Musician Computer Interaction, in Music Department2011, Faculty of Arts, Humanities and Social Sciences: Belfast. p. 206.

[9] Faria, B.M., et al., Machine Learning Algorithms applied to the Classification of Robotic Soccer Formations ans Opponent Teams, in IEEE Conference on Cybernetics and Intelligent Systems (CIS)2010: Singapore. p. 344 - 349

[10] Mannini, A. and A.M. Sabatini, Machine learning methods for classifying human physical activity from on-body accelerometers. Sensors, 2010. 10(2): p. 1154-75.

[11] Maldonado-Báscon, S., et al., Road-Sign detection and Recognition Based on Support Vector Machines, in IEEE Transactions on Intelligent Transportation Systems2007. p. 264-278.

[12] Vicen-Bueno, R., et al., Complexity Reduction in Neural Networks Appplied to Traffic Sign Recognition Tasks, 2004.

[13] Witten, I.H., E. Frank, and M.A. Hall, Data Mining - Pratical Machine Learning Tools and Techniques. Third Edition ed2011: Elsevier.

[14] Snyder, W.E. and H. Qi, Machine Vision2004: Cambridge University Press.

[15] Stephan, J.J. and S. Khudayer, Gesture Recognition for Human-Computer Interaction (HCI). International Journal of Advancements in Computing Technology, 2010. 2(4): p. 30-35.

[16] Ben-Hur, A. and J. Weston, A User's Guide to Support Vector Machines, in Data Mining Techniques for the Life Sciences2008.

[17] Ke, W., et al., Real-Time Hand Gesture Recognition for Service Robot. 2010: p. 976-979.

[18] Camastra, F. and A. Vinciarelli, Machine Learning for Audio, Image and Video Analysis2008: Springer.

[19] Alpaydin, E., Introduction to Machine Learning2004: MIT Press.

[20] Herbrich, R., Learning Kernel Classifiers2003: MIT Press.

[21] Dayan, P., Unsupervised Learning, in The MIT Encyclopedia of the Cognitive Sciences, MIT, Editor.

[22] Davies, E.R., Machine Vision - Theory, Algorithms, Practicalities. 3rd Edition ed, ed. Elsevier2005: Morgan Kaufmann.

[23] Roth, M., et al., Computer Vision for Interactive Computer Graphics, in IEEE Computer Graphics And Applications1998, Mitsubishi Electric Research Laboratory. p. 42-53.

[24] Freeman, W.T. and M. Roth, Orientation Histograms for Hand Gesture Recognition, 1994, MITSUBISHI ELECTRIC RESEARCH LABORATORIES, CAMBRIDGE RESEARCH CENTER.

[25] Fitzgibbon, A.W., Hand Gesture Recognition Using Computer Vision, University of Oxford.