

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/242316687>

Multiclass Hierarchical SVM for Recognition of Printed Tamil Characters

Article · January 2007

CITATIONS

20

READS

112

5 authors, including:



Shivsubramani Krishnamoorthy

Amrita Vishwa Vidyapeetham

23 PUBLICATIONS 191 CITATIONS

[SEE PROFILE](#)



Ajay Vadakkepatt

Qualcomm

19 PUBLICATIONS 345 CITATIONS

[SEE PROFILE](#)



Soman Kp

Amrita Vishwa Vidyapeetham

665 PUBLICATIONS 4,560 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



Deep Learning for Network Flow Analysis and Malware Classification [View project](#)



Stock Price Prediction Using Deep Learning Models [View project](#)

Multiclass Hierarchical SVM for Recognition of Printed Tamil Characters

Shivsubramani K, Loganathan R, Srinivasan CJ, Ajay V, Soman KP

Centre for Excellence in Computational Engineering

Amrita Vishwa Vidyapeetham

Tamilnadu, India

{ramanand, r_logu, cj_srinivasan, v_ajay}@ettimadai.amrita.edu

Abstract

This paper presents an efficient method for recognizing printed Tamil characters exploring the inter-class relationship between them. This is accomplished using Multiclass Hierarchical Support Vector Machines [Crammer *et al.*, 2001; Weston *et al.*, 1998], a new variant of Multi Class Support Vector Machine which constructs a hyperplane that separates each class of data from other classes. 126 unique characters in Tamil language have been identified. A lot of inter-class dependencies were found in them based on their shapes. This enabled the characters to be organized into hierarchies thereby enhancing the process of recognizing the characters. The System was trained using features extracted from the binary character sub-images of sample documents using Hu's [Hu., 1962; Jain *et al.*, 1996] moment invariant feature extraction method. The system fetched us promising results in comparison with other classifying algorithms like KNN, Bayesian Classifier and decision trees. An accuracy of 96.85% was obtained in the experiments using Multiclass Hierarchical SVM

1 Introduction

Support vector machine [Burges, 1998; Cristianini *et al.*, 2000] is a training algorithm for learning, classification and regression rules from data. It is emerging as a very efficient learning methodology in Artificial Intelligence. Fundamentally SVMs are binary classification algorithm with a strong theoretical foundation in statistical learning theory [Vapnik, 1998]. Their ease of use, theoretical appeal, and remarkable performance has made them the system of choice for many learning problems. It is noted by [Cristianini *et al.*, 2002] that the limitations in most of the non-SVM learning algorithms proposed in the past 20 years had been due to the fact that they were based, to a large extent, on heuristics or on loose analogies with natural learning systems. The new pattern-recognition SVM algorithms overcome such limitations with a strong underlying mathematical foundation.

The most crucial stage in the process of Optical Character Recognition (OCR) [Nagy, 1992] is that of recognizing the

characters and classifying them. The other processes involved include preprocessing activities like binarization and skew estimations. It is followed by major phases like Segmentation and Feature Extraction.

Every character in a language forms a class. Character recognition, thus, involves classification of characters into multi-classes. Of the 126 unique characters identified in Tamil language, inter-class dependencies were found within many characters due to the similarity in their shapes. This enabled them to be organized into hierarchies, thus enhancing and simplifying the process classification. Taking advantage of the inter-class dependencies within the character a hierarchical based classification is possible based on the views put forth by [Szedmak *et al.*, 2005]. Combining both the views together, a Multiclass Hierarchical SVM algorithm was devised and is understood to be very efficient methods for character classification.

Experimental outcome of the algorithm fetched us better results compared to other classifiers like Multilayer perceptron, KNN, Naive Bayes, decision tree and other rule based classifiers. The paper presents a detailed comparative study of the efficiency of the various classifiers.

2 Multiclass formulation of the SVM with vector output

We will now see how the multiclass formulation and interpretation differs from classical binary SVMs. First, class labels are vectors instead of +1s and -1s in the binary SVM. Thus class labels in binary SVM belong to one dimensional subspace where as for Multiclass SVM class label belongs to multi-dimensional subspace. Second, \mathbf{W} that defines the separating hyper plane in Binary SVM is a vector. In Multiclass, \mathbf{W} is a Matrix. We can imagine the job of \mathbf{W} in two-class SVMs is to map the data/feature vector into one-dimensional subspace. In multiclass SVM, the natural extension is then, mapping data/feature space into vector label space whose defining bases are vectors. In other words multiclass learning may be viewed as vector labeled learning or vector value-learning.

Assume we have a sample S of pairs $\{(y_i, x_i) : y_i \in H_y, x_i \in H_x, i = 1, \dots, m\}$ independently and identically generated by an unknown multivariate distribution P . The Support Vector Machine with vector output is

realized on this sample by the following optimization problem.

$$\min \frac{1}{2} \text{tr}(\mathbf{W}^T \mathbf{W}) + C e^T \xi$$

Subject to

$$\{\mathbf{W} | \mathbf{W}: H_{\phi(x)} \rightarrow H_y, \mathbf{W} \text{ is a linear operator}\},$$

$$\{\mathbf{b} | \mathbf{b} \in H_y\}, \text{ bias vector}$$

$$\{\xi | \xi \in H_m\}, \text{ slack or error vector}$$

$$\langle \mathbf{y}_i, (\mathbf{W}\phi(\mathbf{x}_i) + \mathbf{b}) \rangle \geq q_i - p_i \xi_i, i = 1, \dots, m,$$

$$\xi \geq 0 \quad (1)$$

where $\mathbf{0}$ denote the vectors with components 0. The real values q_i and p_i denote normalization constraints that can be chosen from the set of values $\{1, \|\mathbf{y}_i\|, \|\phi(\mathbf{x}_i)\|, \|\mathbf{y}_i\| \|\phi(\mathbf{x}_i)\|\}$ depending on the particular task. The bias term \mathbf{b} can be put as zero because it has been shown in [Kecman *et al.*, 2005] that polynomial and RBF kernel do not require the bias term. To understand the geometry of the problem better, first we let q_i and p_i be 1, then the magnitude of the error measured by the slack variables ξ_i will be the same independently of the norm of the feature vectors. Introducing dual variables $\{\alpha_i | i = 1, \dots, m\}$ to the margin constraints and based on the Karush-Kuhn-Tucker theory we can express the linear operator \mathbf{W} by using the tensor products of the output and the feature vectors, that is

$$\mathbf{W} = \sum_{i=1}^m \alpha_i \mathbf{y}_i \phi(\mathbf{x}_i)^T \quad (2)$$

The dual gives

$$\min \sum_{i,j=1}^m \alpha_i \alpha_j \overbrace{\langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle}^{K^\phi_{ij}} \overbrace{\langle \mathbf{y}_i, \mathbf{y}_j \rangle}^{K^y_{ij}} - \sum_{i=1}^m \alpha_i, \quad (3)$$

Subject to

$$\{\alpha_i | \alpha_i \in \mathbb{R}\},$$

$$\sum_{i=1}^m (\mathbf{y}_i)_t \alpha_i = 0, t = 1, \dots, \dim(H_y),$$

$$C \geq \alpha_i \geq 0, i = 1, \dots, m,$$

where we write the output of inner products in the objective as kernel items $\langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle \langle \mathbf{y}_i, \mathbf{y}_j \rangle = K^\phi_{ij} K^y_{ij}$ where stand for the elements of the kernel matrices for the feature vectors and for the label vectors respectively. Hence, the vector labels are kernelized as well. The synthesized kernel is the element-wise product of the input and the output kernels, an operation that preserves positive semi-definiteness. The main point to be noted in the above formulation (1) is the constraint equations.

$$\langle \mathbf{y}_i, (\mathbf{W}\phi(\mathbf{x}_i)) \rangle \geq 1 - \xi_i, i = 1, \dots, m,$$

$$\xi_i \geq 0, i = 1, \dots, m,$$

Here when we project $\mathbf{W}\phi(\mathbf{x}_i)$ onto \mathbf{y}_i , we are restricting the resulting value to be always less than or equal to one. There seems to be no compelling reason for such a restriction. So if we allow $\langle \mathbf{y}_i, (\mathbf{W}\phi(\mathbf{x}_i)) \rangle$ to take value around 1 (both sides) the nonnegative restriction on ξ_i goes out. Further the inequality constraint given below becomes equality constraint

$$\langle \mathbf{y}_i, (\mathbf{W}\phi(\mathbf{x}_i)) \rangle \geq 1 - \xi_i, i = 1, \dots, m,$$

$$\langle \mathbf{y}_i, (\mathbf{W}\phi(\mathbf{x}_i)) \rangle = 1 - \xi_i, i = 1, \dots, m,$$

The above change in turn necessitates 1-norm minimization term $C e^T \xi$ in the objective function to take the two-norm form $\frac{1}{2} C \xi^T \xi$. The formulation given below (4) basically can be thought of an extension of the formulation given [Mangasarian *et al.*, 2001] for a two-class SVM.

$$\min \frac{1}{2} \text{tr}(\mathbf{W}^T \mathbf{W}) + \frac{1}{2} C \xi^T \xi$$

Subject to

$$\{\mathbf{W} | \mathbf{W}: H_{\phi(\mathbf{x})} \rightarrow H_y, \mathbf{W} \text{ linear operator}\},$$

$$\{\xi | \xi \in H_m\}, \text{ slack or error vector}$$

$$\langle \mathbf{y}_i, (\mathbf{W}\phi(\mathbf{x}_i)) \rangle = 1 - \xi_i, i = 1, \dots, m, \quad (4)$$

Lagrangian is given by,

$$L = \frac{1}{2} \text{tr}(\mathbf{W}^T \mathbf{W}) + \frac{1}{2} C \xi^T \xi - \sum_{i=1}^m \alpha_i (\mathbf{y}_i^T \mathbf{W} \phi(\mathbf{x}_i) - 1 + \xi_i)$$

Solving for primal variables,

$$\begin{aligned} \frac{\partial L}{\partial \mathbf{W}} &= \mathbf{W} - \sum_{i=1}^m \alpha_i \mathbf{y}_i \phi(\mathbf{x})^T = \mathbf{0} \\ \mathbf{W} &= \sum_{i=1}^m \alpha_i \mathbf{y}_i \phi(\mathbf{x})^T \end{aligned} \quad (5)$$

$$\begin{aligned} \frac{\partial L}{\partial \xi_i} &= C \xi_i - \alpha_i = 0 \quad , \\ \xi_i &= \frac{\alpha_i}{C} \\ \xi &= \frac{\alpha}{C} \end{aligned} \quad (6)$$

Substituting $\mathbf{W} = \sum_{i=1}^m \alpha_i \mathbf{y}_i \phi(\mathbf{x})^T$ and $\xi = \frac{\alpha}{C}$ in the constraint (1), we obtain

$$\begin{aligned} (\mathbf{K}^y \cdot * \mathbf{K}^\phi) \alpha &= \mathbf{e} \cdot \frac{\alpha}{C} \\ \mathbf{K} \alpha &= \mathbf{e} \cdot \frac{\alpha}{C} \end{aligned}$$

Where $\mathbf{K} = \mathbf{K}^y \cdot * \mathbf{K}^\phi$

$$\begin{aligned} (\mathbf{K} + \frac{\mathbf{I}}{C}) \alpha &= \mathbf{e} \\ \mathbf{Q} \alpha &= \mathbf{e} \end{aligned}$$

$$\begin{aligned} \text{Where } \mathbf{Q} &= (\mathbf{K} + \frac{\mathbf{I}}{C}) \\ \alpha &= \mathbf{Q}^{-1} \mathbf{e} \end{aligned} \quad (7)$$

This leads to a closed form solution for SVM Training. Here α 's are unrestricted in sign and unbounded.

3 Multiclass classification

The multiclass classification can be implemented within the framework of the vector valued SVM. Let us assume the label vectors are chosen out of a finite set $\{\hat{\mathbf{y}}_1, \dots, \hat{\mathbf{y}}_T\}$ in the learning task. The decision function predicting one of these labels can be expressed by using the predicted vector output [Shawe-Taylor *et al.*, 2005]

$$d(x) = \arg \max_{t=1, \dots, T} \sum_{i=1}^m \alpha_i K^y(\hat{\mathbf{y}}_t, \mathbf{y}_i) K^\phi(\mathbf{x}_i, \mathbf{x}) \quad (8)$$

Now we are able to set up a multiclass classification. In [Shawe-Taylor *et al.*, 2005] following two rules for setting up labels are given.

$$(\mathbf{y}_i)_t = \begin{cases} 1 & \text{if } i \text{ belongs to category } t, t=1, \dots, T \\ 0 & \text{otherwise.} \end{cases} \quad (9)$$

$$(\mathbf{y}_i)_t = \begin{cases} \sqrt{\frac{T-1}{T}} & \text{if } i \text{ belongs to } t, t=1, \dots, T \\ -\frac{1}{\sqrt{T(T-1)}} & \text{otherwise} \end{cases} \quad (10)$$

4 Hierarchical based learning

Introducing the concept of hierarchical organizing of the characters, the above mentioned equation (10) can be represented as equation (11). The hierarchy is conceptualized manually wherein every child node holds no relationship or similarity with nodes other than its siblings alone. The hierarchy learning [Szedmak *et al.*, 2005] is realized via an embedding of each path going from a node to the root of the tree. Let V be the set of nodes in the tree. A path $p(v) \subset V$ is defined as a shortest path from the node v to the root of the tree and its length is equal to $|p(v)|$. The set $I = 1, \dots, |V|$ gives an indexing of the nodes. The embedding is realized by a vector valued function $\varphi: V \rightarrow \mathbb{R}^{|V|}$, and the components of $\varphi(v)$ are given by

$$\varphi(v)_i = \begin{cases} r & \text{if } v_i \notin p(v), \\ sq^k & \text{if } v_i \in p(v) \text{ and } k = |p(v)| - |p(v_i)| \end{cases} \quad (11)$$

Where r, q, s are the parameters of embedding. The parameter q can express the diminishing weight of the nodes being closer to the root. If $q=0$, assuming $0^0 = 1$, then the intermediate nodes and the root are discarded, thus we have a multiclass classification problem. The value of r can be 0 but some experiments show it may help to improve the classification performance. This method was successfully applied to WIPO-alpha patent dataset and Reuters Corpus and were known to give good results against them.

5 Experiments

Tamil is a South Indian Language mainly spoken in southern parts of India, Sri Lanka, Malaysia and Singapore. Tamil character set contains 12 vowels, 18 consonants and totally 247 alphabets. 126 unique commonly occurring characters in shape have been identified. Hierarchy is built based on the 126 characters as explained in the section 5.4. Thus the classification was to be done into 126 classes. The following flow chart Figure 1 describes the steps involved in preparing the features extracted for training the system and for classification thereby.

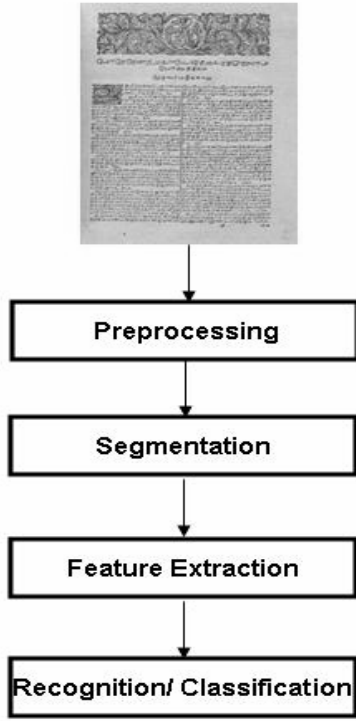


Figure 1: OCR process

5.1 Preprocessing

Documents were digitized and stored as gray scale Bitmap images. Binarization was performed based on a threshold value applied on the image. Since the scanned images were noise free to a considerable extent, a noise reduction technique was not required to be performed on the image.

5.2 Segmentation

Segmentation was performed in two phases. (i) Line segmentation wherein each line in the document was segmented using horizontal profile. (ii) Character segmentation wherein each character in the line was segmented using 8-connected component analysis [Haralick *et al.*, 1992]. The two phase approach was adopted based on a comparative study where this approach yielded a better result.

5.3 Feature extraction

Feature extraction involves extracting the attributes that best describe the segmented character image. Moment based invariants are the most commonly used feature extraction method in many applications. It explores information across an entire image and it can capture some of the global properties like the overall image orientation. Hu's [Hu., 1962; Jain *et al.*, 1996] moment invariant method was adopted since it is invariant to scaling, rotation and image translation. Seven invariant features were extracted based on the following non linear functions,

$$\phi_1 = \eta_{20} + \eta_{02}$$

$$\phi_2 = (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2$$

$$\phi_3 = (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2$$

$$\phi_4 = (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2$$

$$\phi_5 = (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2]$$

$$\phi_6 = (\eta_{20} - \eta_{02})[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03})$$

$$\phi_7 = (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2]$$

Where η_{pq} is $(p + q)^{th}$ order normalized central moment.

5.4 Hierarchical labeling

A lot of inter class dependencies were found in Tamil characters based on their shapes. Many characters exhibit a lot of similarity with other characters. The feature values of such a pair of characters have a very minimal difference. Some examples of such character pairs are as shown in Figure 2.

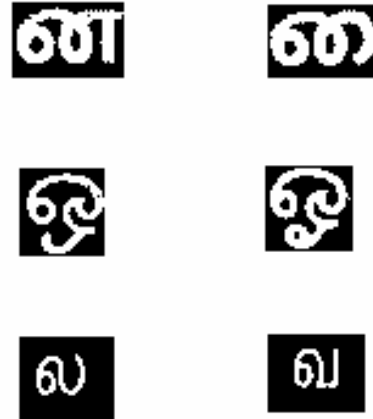


Figure 2: Similarity in shapes

Taking advantage of such a property exhibited by the Tamil characters, the characters exhibiting similarity were organized into hierarchies which eased the process of classification. Figure 3 shows a hierarchical tree structure of some selected characters with similarity in shapes.

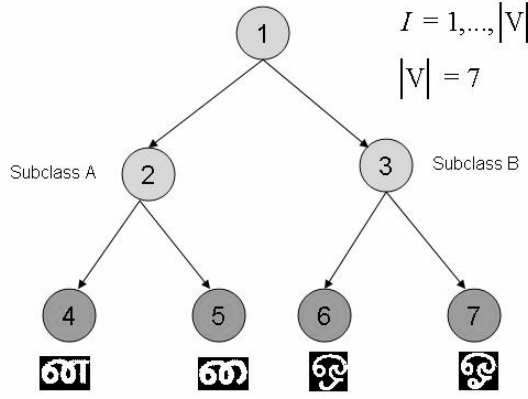


Figure 3: Hierarchical labeling

Incase the character feature exhibits the property of a particular subclass, the classes not belonging to this hierarchy need not be considered for classification at all. Thereby a 126 class classification problem can be broken down to a 10 class or 8 class problem. This, to a large extent, enhances the accuracy and efficiency by increasing the inter-class differentiability factor.

5.5 Training

Training data set was generated by labeling the features extracted from the test character image, with the corresponding class. A training dataset for a particular class, on average, contains 20 sample training data.

6 Results

Multiclass Hierarchical SVM turned out to be a very efficient method in process of classification. The accuracy of the algorithm depended on two parameter settings (RBF Kernel parameter σ and regularization parameter C). The form of RBF kernel used is $e^{-\sigma(\|x_1 - x_2\|)^2}$. The system had to be fine tuned on these values in order to obtain a better accuracy. Figure 4 shows the improving performance/accuracy of the system with the changing values of the parameters.

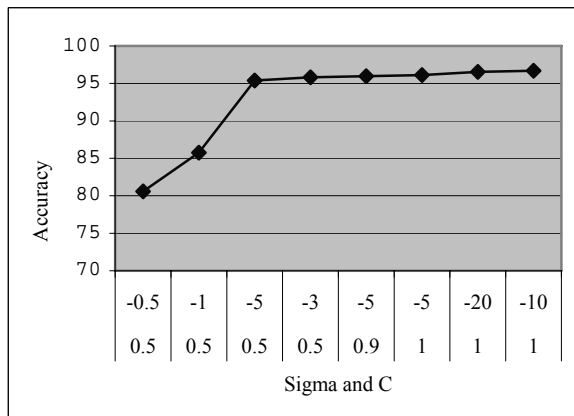


Figure 4: SVM Classifier Accuracy

The accuracy rate yielded by the SVM classifier was quite commendable. Based on a comparative study performed, Multiclass Hierarchical SVM showed better accuracy rate than many other classifiers used like Multilayer perceptron, KNN, Naive Bayes, decision tree and other rule based classifiers.

The system was tested thrice using 3, 7 and 20 most similar characters respectively. The accuracy rate was calculated using 10 fold cross validation technique. Table 1 depicts the comparison values of the accuracy yielded by the various classifiers.

| Classifier | Accuracy (%) with | | |
|-----------------------------|-------------------|---------------|----------------|
| | 3 char-acters | 7 char-acters | 20 char-acters |
| Multiclass Hierarchical SVM | 96.85 | 96.23 | 96.86 |
| Multilayer Perceptron | 91.8 | 95.45 | 93.43 |
| KNN | 89.40 | 90.05 | 89.90 |
| Naïve Bayes | 84.5 | 88.90 | 88.20 |
| Decision Trees | 91.0 | 92.84 | 93.23 |

Table 1: Comparison of classifier performances

7 Enhancements

- The system can be enhanced by including a module of Language heuristics wherein a character not leading to a particular class could be classified based on predictions made using the general language grammar and rules.
- Another enhancement possible is compiling a language dictionary wherein in a situation of ambiguity, classification could be performed based on language semantics
- Apart from the 126 characters identified here, there exist some ancient Tamil characters that are not used commonly. Such characters can also be taken into consideration for a broader aspect.
- The system can also be extended to other oriental languages. We are planning to work on some Indian languages like Sanskrit, Hindi, and Malayalam etc., which exhibit the property of similarity between characters in shapes and can be organized into hierarchies.

8 Conclusion

The paper presented an efficient algorithm for classification of characters using Multiclass Hierarchical SVM, a variant of Multiclass SVM. The system was applied for the recognition of printed Tamil language characters. The experimental procedures are explained and the results listed out depicting the efficiency of the system. The algorithm did prove more efficient than some of the commonly used classifiers. Some merits of our algorithm are:

- Strong mathematical model foundation rather than heuristics and analogies.
- Efficient in terms of accuracy in comparison with many commonly used classifiers

References

- [Burges, 1998] C.J.C. Burges, A Tutorial on Support Vector Machines for Pattern Recognition,” *Data Mining and Knowledge Discovery*, 1998
- [Chen., 2003] Qing Chen, Evaluation of OCR Algorithms for Images with Different Spatial Resolutions and Noises, Master thesis, School of Information Technology and Engineering, University of Ottawa, 2003
- [Crammer *et al.*, 2001] Koby Crammer and Yoram Singer, On the Algorithmic Implementation of Multiclass Kernel-based Vector Machines, *Journal of Machine Learning Research* 2 (2001), pp. 265-292
- [Cristianini *et al.*, 2000] Nello Cristianini, John Shawe-Taylor, An Introduction to Support Vector Machines and Other Kernel-based Learning Methods, Cambridge University Press; 1st edition, 2000
- [Cristianini *et al.*, 2002] Nello Cristianini, Bernhard Scholkopf, Support vector machines and kernel methods: the new generation of learning machines, *Articles AI Magazine*, Fall, 2002
- [Joachims, 1998] T. Joachims, Making Large-Scale SVM Learning Practical, *Advances in Kernel methods-Support Vector Learning*, MIT Press, 1998.
- [Haralick *et al.*, 1992] Haralick, Robert M., and Linda G. Shapiro, Computer and Robot Vision, Volume I, Addison-Wesley, 1992, pp. 28-48.
- [Hu., 1962] MK Hu, Visual Pattern Recognition by Moment Invariants, *IRE Trans. Information Theory*, vol. 8, pp. 179-187, 1962
- [Jain *et al.*, 1996] O. D. Trier, A. K. Jain and T. Taxt, Feature extraction methods for character recognition - A survey, *Pattern Recognition* 29, pp. 641-662, 1996.
- [Kecman *et al.*, 2005] V. Kecman, T.M. Huang and M. Vogt, Iterative Single Data Algorithm for Training Kernel Machines from Huge Data Sets: Theory and Performance, *Support Vector Machines: Theory and Applications*, Springer-Verlag, Studies in Fuzziness and Soft Computing, Vol. 177, Chap. 12, 2005, pp. 255-274
- [Mangasarian *et al.*, 2001] G. Fung and O.L. Mangasarian, Proximal Support Vector Machine Classifiers, *KDD 2001: Seventh ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, San Francisco August 26-29, 2001
- [Nagy, 1992] G. Nagy, On the Frontiers of OCR, *Proceedings of the IEEE*, vol. 40, #8, pp. 1093-1100, July 1992.
- [Osuna *et al.*, 1997] E. Osuna, R. Freund, and F. Girosi, “An Improved Training Algorithm for Support Vector Machines,” *Proc. IEEE Neural Networks for Signal Processing VII Workshop*, IEEE Press, Piscataway, N.J., 1997, pp.276–285.
- [Schölkopf, *et al.*, 2002] B. Schölkopf and A. J. Smola. *Learning with Kernels*. MIT Press, 2002.
- [Shawe-Taylor *et al.*, 2005] Sandor Szedmak and J. Shawe-Taylor, Multiclass Learning at One-class Complexity, Technical Report, ISIS Group, Electronics and Computer Science, 2005.
- [Szedmak *et al.*, 2005] Sandor Szedmak, John Shawe-Taylor, Learning Hierarchies at Two-class Complexity, *Kernel Methods and Structured domains*, NIPS 2005
- [Vapnik, 1995] V. Vapnik, The Nature of Statistical Learning Theory, Springer-Verlag, New York, 1995.
- [Vapnik, 1998] V.Vapnik, Statistical Learning Theory. John-Wiley and Sons, Inc., New York, 1998
- [Weston *et al.*, 1998] J. Weston, C. Watkins. Multi-class support vector machines. Technical Report CSD-TR-98-04, Department of Computer Science, Royal Holloway, University of London, Egham, TW20 0EX, UK, 1998.