SVM with Stochastic Parameter Selection for Bovine Leather Defect Classification

Roberto Viana[1], Ricardo B. Rodrigues[1], Marco A. Alvarez[2], and Hemerson Pistori[1]

[1] GPEC - Dom Bosco Catholic University,
Av. Tamandare, 6000 Campo Grande, Brazil
http://www.gpec.ucdb.br
[2] Department of Computer Science
Utah State University
Logan, UT 84322-4205, USA
{roberto,ricardo}@acad.ucdb.br,
pistori@ucdb.br,
malvarez@cc.usu.edu

Abstract. The performance of Support Vector Machines, as many other machine learning algorithms, is very sensitive to parameter tuning, mainly in real world problems. In this paper, two well known and widely used SVM implementations, Weka SMO and LIBSVM, were compared using Simulated Annealing as a parameter tuner. This approach increased significantly the classification accuracy over the Weka SMO and LIBSVM standard configuration. The paper also presents an empirical evaluation of SVM against AdaBoost and MLP, for solving the leather defect classification problem. The results obtained are very promising in successfully discriminating leather defects, with the highest overall accuracy, of 99.59%, being achieved by LIBSVM tuned with Simulated Annealing.

Keywords: Support Vector Machines, Pattern Recognition, Parameter Tuning.

1 Introduction

The bovine productive chain plays an important role in the Brazilian economy and it has been considered as the owner of the largest cattle herd in the world [1]. However, according to [2] only 8.5% of Brazilian leather achieves high quality. Recently, the Brazilian Agricultural Research Corporation (EMBRAPA) suggested the pursuit of automation for improving the reliability of the national grading system for bovine raw hide[1]. In particular, the authors of this paper believe that designing computational systems for the automatic classification of leather defects represents a relevant contribution to the government and industrial needs.

---

[1] Normative instruction number 12, December 18th, 2002, Brazilian Ministry of Agriculture, Livestock and Food Supply.

The defect classification in materials like, wood, metals, woven and leathers is reported to be made visually in [3]. In general, this task involves the product surface analysis in order to identify fails. Recalling that such task requires laborious and precise work, it is very common to face the occurrence of errors during the analysis.

The visual inspection of leather surfaces for analysis of defects can be modeled using computer vision techniques as reported in [3,4,5,6,7,8,9]. Nonetheless, leather is considered a complex object for analysis since it can present a large range of differences in color, thickness, wrinkledness, texture and brightness [6].

In order to address the automatic classification of leather defects, this paper proposes the use of computer vision techniques and machine learning algorithms. This work is part of the DTCOURO project[2] which proposes the development of a completely automated system, based on computer vision, for bovine raw hide and leather classification and grading.

Among the existing supervised learning algorithms, Support Vector Machines (SVM) have been widely used for classification denoting great generalization power and capacity for handling high-dimensional data [10]. However, despite its success, SVMs still are very sensitive to the definition of initial parameters. Determining the right parameter set is often computationally expensive and over-fitting may occur when the training set does not contain a sufficient number of training examples. Furthermore, the parameter selection has a preponderant effect on the effectiveness of the model.

Having in mind these considerations, the approach proposed and evaluated in this paper consists of the following contributions:

1. The use of Interaction Maps [11], Co-occurrence Matrices [12], RGB and the HSB Color Space for extracting texture and color features from a given set of raw hide leather images. The proposed methods are based on the feature extraction algorithms experimented in [9];
2. An empirical evaluation of the use of a selected supervised learning algorithms set for solving the leather defect classification problem. Two different implementations of SVMs (LIBSVM[3] and SMO [13]) in conjunction with a stochastic approach, namely simulated annealing, for SVM parameter tuning were posted against, Multilayer Perceptron (MLP) and an adaptive boosting of decision trees and K-NN using the well-known AdaBoost [14] algorithm. The results obtained are very promising in successfully discriminating leather defects. The highest overall accuracy achieved by SVM is 99.59%.

The remaining of this paper is organized as follows: Section 2 introduces concepts and previous work related to the leather inspection using computer vision and automatic classification of leather defects. Section 3 gives an overview of the selected machine learning algorithms. The experimental settings and results are presented in Section 4. Finally, conclusions and research directions are given in Section 5.

---

[2] `http://www.gpec.ucdb.br/dtcouro`

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

2   Related Work

The discussion of related work presented here is divided into two main parts. First, related work on leather defect detection and classification is discussed, and after that, an overview on the use of stochastic methods for SVM parameter optimization is shown. Bear in mind that this sections is not intended to present an exhaustive literature review on related work, in its place, the most suggestive topics will be covered.

High quality leather is very important in numerous industrial segments. The good appearance of products made using leather depends on the absence of defects in its surface. Bovine leather, in particular, is characterized by the emergence of defects when the animal is still alive and it goes until the tanning process. Defects are mostly provoked by: 1) wounds during the productive phase (e.g. cuts, fighting with other males, brand marks using hot iron, infections, among others); 2) exposure of cattle to ectoparasites and inadequate management [15]; and 3) development of problems during transportation and conservations phases. Defects during tanning and post-processing are much less common, as they are controlled by the tanneries, which have in the leather quality their main business. For a more detailed description of potential causes for common leather defects in the Brazilian leather productive chain the reader can refer to [16].

Roughly speaking, leather defects can be observed in raw hide, the untanned hide of a cattle, or in wet blue leather which is a hide that has been tanned using chrominus sulphate. Wet blue leather is an intermediate stage between untanned and finished leather. The reader can examine Figure 1 to have a clue about the appearance of raw hide and wet blue leather. In general, the detection and classification of leather defects is conducted on wet blue leather, because even without defects, bovine raw hide has a very complex surface.
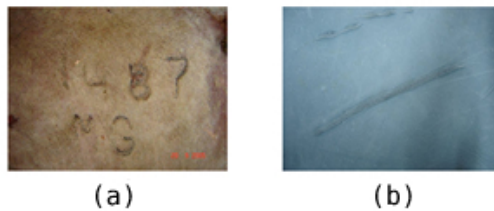

Fig. 1. (a) Image of a 'brand mark' defect on bovine raw hide taken after skinning and before tanning. (b) Image of a 'open scar' defect on bovine wet blue leather during the first stage of the tanning process.

Yeh and Perng in [3] propose and evaluate semi-automatic methods for wet blue leather defects extraction and classification. Their results are reliable and effective, when compared with human specialists. The main contribution of the work is a fully quantified grading system, called *demerit count reference standard for leather raw hides*, but the authors also point out that one of the drawbacks of

their proposal is the need for human, specialized intervention, for counting the total number of demerits on a wet blue leather. A leather inspection method, based on Haar's wavelets, is presented by Sobral in [4]. The system is reported to perform in real time, at the same level of an experienced operator [4] and to outperform previous methods, based on Gabor filters, like the one described in Kumar and Pang [5]. Although not clearly stated in Sobral's paper, the system seems to have been experimented only on finished leather, a much simpler problem than raw hide or wet blue leather defect extraction. A dissimilarity measure based on $\chi 2$ criteria has been used to compare gray-level histograms from sliding windows (65x65 pixels) of a wet blue leather image to an averaged histogram of non-defective samples in [6]. The results of the $\chi 2$ test and an experimentally chosen threshold are used to segment defective regions of the leather. The approach has not been used to identify the defect type. The segmentation of defective regions from wet blue leather images, using histogram and co-occurrence based features, has been investigated in [7].

On the other hand, the approach proposed in this paper considers the use of SVMs for defect classification on raw hide leather regions. Nevertheless, SVMs achieve good performance only with appropriate parameter estimation, specifically its C parameter which denotes the penalty on misclassified examples and the γ parameter, necessary for the RBF kernel, the one chosen for the experiments.

Proposals for estimating the free parameters range from manual setting by experts with *a priori* knowledge on the data to the computationally expensive grid search. When using a grid search, a number of combinations of SVM parameters C and γ are tried within a predefined range. Empirical estimation is undesired because it does not provide any guarantee on selecting the best parameters, on the other hand, the precision of grid search depends on the range and granularity chosen. One would be interested in an automatic parameter selection method without traversing the entire parameters space. Stochastic heuristics guided by the simulated annealing [17] algorithm are very suitable for selecting the best parameters for a given training set. Formerly this approach has been presented in [18,19].

Motivated by the outstanding performance of SVMs in solving classification tasks, the goal of the experiments conducted in the present work is to validate their use in a real problem such as the defect classification in raw ride leather. Moreover, simulated annealing is applied for parameter selection in order to conduct a global optimization on the SVM parameters C and γ.

3   Supervised Learning Approach for Defect Classification

Machine Learning techniques during the last years have been successfully used to solve significant real world applications [20]. In this paper, the authors propose the use of machine learning for solving the defect classification problem in raw ride leather images, where a classifier is trained to learn the mapping function between a set of features describing a particular region of a given image and the type of the leather defect. This is the central point on the approach proposed here.

The goal of this section is to present how the defect classification problem can be modeled as a supervised learning problem. For this purpose, initially basic definitions are presented, followed by an overview of the selected learning algorithms.

3.1   Basic Definitions

Regarding the supervised learning terminology, the following definitions will be considered in the context of the defect classification problem. A labeled instance is a pair $(\bar{x}, y)$ where $\bar{x}$ is a vector in the d-dimensional space X. The vector $\bar{x}$ represents the feature vector with $d = 145$ attributes extracted from a region within a given raw hide leather image and y is the class label associated with $\bar{x}$ for a given instance, details on the attribute extraction phase are found in section 4.1. Therefore, a classifier is a mapping function from X to Y. The classifier is induced through a training process from an input dataset which contains a number n of labeled examples $(\bar{x}_i, y_i)$ for $1 \leq i \leq n$.

For the experiments, a set of four types of defect has been chosen containing the following elements: tick marks, brand marks made from hot iron, cuts and scabies. These defects have been chosen because they are very common in Brazilian leather. From each region extracted from the raw ride leather images, a set of features were extracted using color and texture attributes.

Applying machine learning algorithms raises the question of how to select the right learning algorithm to use. As stated in previous sections, SVMs with stochastic parameter selection will be experimented and compared according to their effectiveness with the MLPs and the boosting of Decision Trees and K-NN using the Adaboost [14] algorithm.

3.2   Support Vector Machine

Support Vector Machines (SVM), created by Vapnik [21], have become one of the most popular classification algorithms. SVMs are classifiers based on the maximum margin between classes. By maximizing the separation of classes in the feature space, it is expected to improve the generalization capability of the classifiers, which are conceived, in the basic approach, as linear classifiers that split the input data into two classes using a separating hyperplane. The reader can refer to Figure 2 for an illustration of the basic SVM for linearly separable data.

SVMs can also work with non-linearly separable datasets either by mapping the input feature space into higher dimensions using kernel functions or relaxing the separability constraints. In the former it is expected that the same dataset become linearly separable in the higher space whereas in the latter some margin failures are allowed but penalized using the cost parameter C. In fact, this parameter in conjunction with the kernel parameters are critical to the performance of the classifier. In this paper, these parameters are estimated using Simulated Annealing [17], which is a stochastic algorithm for the global optimization problem. The goal is to locate a good approximation to the global optimum for the
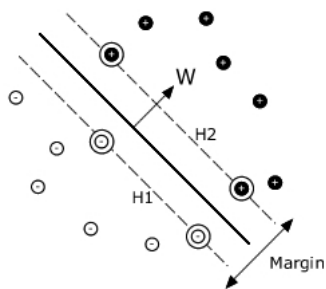
Fig. 2. Illustration of the hyperplane separating two classes. Note the maximum margin separation for the given input examples. SVMs are expected to improve generalization by maximizing the margin.

generalization performance in the SVM's free parameters space. At each step the algorithm replaces the current solution with a probabilistic guess on nearby solutions, controlled by a global adaptive parameter T, the temperature.

SVMs are naturally designed for binary classification, however, available implementations like LIBSVM and SMO [13] provide the extension of SVMs for multi-class problems. Several methods have been proposed for multi-class SVMs by combining binary classifiers. A comparison of different methods for multi-class SVMs is presented in [22].

3.3   AdaBoost and MLPs

Boosting is a general way to improve the accuracy of any given learning algorithm. The basic idea behind boosting refers to a general method of producing very accurate predictions by combining moderately inaccurate (weak) classifiers. AdaBoost is an algorithm that calls a given weak learning algorithm repeatedly, where at each step the weights of incorrectly classified examples are increased in order to force the weak learner to focus on the hard examples. The reader can refer to [14] for a detailed description of AdaBoost. The main motivation to the use of a meta-classifier as AdaBoost is given by the fact that many previous papers have shown stellar performance of AdaBoost with several datasets [23]. In fact, Bauer and Kohavi in [23] show a more realistic view of the performance improvement one can expect. After empirical evaluation of selected weak learners, the authors opted for the J48 Decision Tree (DT) algorithm (the Java implementation of C4.5 integrated in Weka[4]) and IBK (a Java implementation of K-NN integrated in Weka).

The C4.5 algorithm splits data by building a decision tree based on attributes from the training set. Basically at each integration the algorithm selects the best attribute based on information gain and splits the data in 2 subsets. In addition, decision trees have the advantage of: 1) DTs are easy to understand and convert into production rules, allowing fast evaluation of test examples, 2) There are no *a priori* assumptions about the nature of the data.

---

[4] `http://www.cs.waikato.ac.nz/ml/weka`

The k-nearest neighbor (K-NN) is one of the most simple algorithms in machine learning algorithms. The classification of data consists in to gather a majority of votes of its neighbors where the most common class among its k nearest neighbors is assigned to it. The neighbours are selected from a set of correct classified samples that are represented as vectors in a multidimensional feature space. Euclidean distance and Manhattan distance can be used as distance measures.

The Multilayer Perceptron (MLP) is basically a set of processing units organized in layers where the number of layers and the number of units in each layer varies according to the problem, the first layer is called input layer, the last layer is the output layer and all layers between them are called hidden layers, the output layer has one unit for each class in the training set. The units on a layer are usually connected to all units in the layer above and below it and have weight values that denote their behavior and are adjusted during the training process. After the training phase, for all data presented at the input layer the network perform calculations until an output is computed at each of the output layers. It is expected that the correct class have the highest output value in the output layer.

4   Empirical Evaluation

Having in mind the goal of evaluating and comparing the results obtained by the selected classifiers, this section describes the details about the experiments conducted for this paper together with an analysis of their results.

4.1   Dataset

In order to create the dataset for experimentation fifteen bovine images from tanned leather in the raw hide stage were selected from the DTCOURO repository. The images have been taken using a five mega-pixel digital camera during technical visits to slaughterhouses and tanneries located in the region of Mato Grosso do Sul, Brazil, by September 2005. For this project, the images were scaled from high resolution images to $600x450$ pixels with the intention of saving time and space. Empirical evidence have shown that there is no loss of effectiveness when using the scaled images. Furthermore, the images have low environmental characteristics variation.

A set of four types of defect has been chosen, namely, tick marks, brand marks from hot iron, cuts and scabies. As the goal of this work is to distinguish between defects, non defect samples were not considered. One sample of each of these defects can be visualized in Figure 3. The defects were manually segmented using a software module from the DTCOURO project. A total of thirty segments have been extracted from the images including examples of the previously cited leather defects. After the manual segmentation of defects, an algorithm implemented in the DTCOURO project was used to extract windows of 20x20 pixels by scanning all the segments. Each window is an example that belongs to either one of the defects class. A total of 14722 20x20 windows were created in this way.
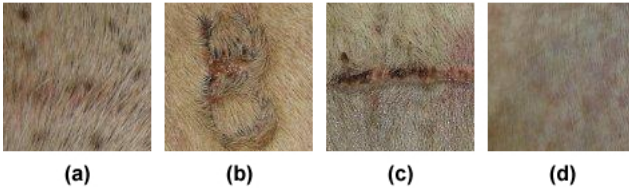
Fig. 3. Sample of (a) tick, (b) brand, (c) cut and (d) scabies over the leathers in raw hide stage

The next step is the feature extraction from each 20x20 window. A set of 139 attributes for each sample were extracted using Interaction Maps [11] and the Grey Level Co-Occurrence Matrices [12] (GLCM) for texture attributes and 6 attributes using the mean values of histograms for hue, saturation and brightness, red, green and blue for color attributes. Interaction Maps is an attribute extraction technique that consists in a directional analysis of the texture. The co-occurrence matrices can be defined over an image as the distribution of gray level pixel values occurring at given offsets [12]. Usually, the values of GLCMs are not directly used as texture features, but some statistics and values calculated from them, like entropy, contrast, angular second moment, inverse difference moment, energy and homogeneity. In this project, the feature extractors are configured based on previous experiments reported in [24], which can be seen in Table 1.

Table 1. Parameters used for the feature extraction techniques used in this project. 139 texture features were extracted from each 20x20 window.

	Int. Maps	Co. Matrices
Initial Angle:	10	10
Final Angle:	180	180
Angle variation:	10	10
Distance (pixels):	2	-
Distance variation:	1	1

For each of the 14722 examples, a feature vector $\bar{x}$ was calculated and stored into the dataset. At the same time, all the training examples were already labeled with one of the following classes: $\{Tick, Brand, Cut, Scabies\}$, the distribution of classes is as follows: 2819 *Tick*, 3716 *Brand*, 2804 *Cut* and 5383 *Scabies* examples, where the number of examples in each class is proportional to the area of each defective region in the original images.

4.2 Experimental Settings

The experiments were conducted using the latest developer version of Weka software [25] and the LIBSVM library written by Chang and Lin [26]. Two different implementations of SVMs, LIBSVM and SMO were tested in conjunction with

MLPs and AdaBoost-J48 and K-NN. For each of the algorithms 5-fold cross-validation was performed over the dataset in order to certify a more reliable estimation of the generalization error [18].

A new module for classifier parameters tuning was developed into DTCOURO project for this work. This module applies parameters recombination that results in significant accuracy improvement. The Simulated Annealing was the algorithm chosen to parameters generation due to its good probability to terminate with the global optimal solution.

LIBSVM is a library for Support Vector Machines developed by Chih-Chung Chang [26] that implements classification, regression and distribution estimation. The implemented classifier solves the SVM quadratic programming problem by decomposing the set of lagrange multipliers. LIBSVM also implements two techniques to reduce the computational time of the evaluation function, shrinking and caching. The technique used to solve multi-class problems is the *one-against-one*.

Sequential Minimal Optimization (SMO) is a SVM training algorithm developed by John C. Platt [13], who claims that SMO is a simple and fast technique to solve the SVM's quadratic problem. The main advantage of SMO compared to other SVM training algorithms is that it always chose the smallest QP problem to solve at each iteration. An other advantage is that SMO does not use storage matrix, which greatly reduces the memory usage.

4.3   Evaluation of Supervised Algorithms

The experiments are basically exploratory and were conducted with the intention of evaluating the effectiveness and efficiency of the algorithms over the leather defect detection. The works in this subsection can be divided in two parts. The first experiment shows the time, best parameters founded and overall accuracy of SVMs tuning. In the second part the results of best tuned SVMs are compared with other well know algorithms and analyzed using traditional measures including, precision, recall, overall accuracy and area under the ROC curve.

SVMs tuning. Initially, the goal of the experiments was the search for the best SVMs C and γ parameters. The classifier parameters tuning module was applied over a smaller subset of the dataset for time saving proposes, where at each iteration the C and γ values were evaluated using 5-fold cross validation. The evaluation measure is the overall accuracy. The initial values for C and γ are the default values suggested by Weka. In Table 2 the reader can notice the execution time, best values for C and γ and their respective overall accuracy (number of correctly classified examples), using the default values (Def. Acc.) and the optimized values (Opt. Acc.).

As the reader can observe, the results clearly show a higher time performance achieved by the LIBSVM implementation. The final overall accuracy in both cases was improved, even though one can conclude that the use of LIBSVM is recommended due to its high performance. Despite the slow computation time of SA it is still practicable in this situation since it needs to be executed only once. One of the possible reasons for the difference in time can be credited to the

Table 2. Running time, best C and γ, default accuracy and accuracy with Simulated Annealing optimization for SVM parameter estimation using the LIBSVM and SMO implementations

	Time	Best C and γ	Def. Acc.	Opt. Acc.
SMO	35655s	24.165 0.931	88.95%	93.10%
LIBSVM	12786s	49.494 1.008	76.16%	99.59%

LIBSVM shrinking and caching implementation. Table 2 also shows that Simulated Annealing optimization increased the classification accuracy in 23% over the standard LIBSVM parameter configuration and 5% over the SMO standard.

Classifiers comparison. The confusion matrix is a $|Y| \times |Y|$ bi-dimensional array where the position (i, j) denotes the number of examples of class i predicted as examples of the class j. Roughly speaking, each column represents the predicted examples and each row represents the actual examples. Such matrix can be used to compare the classifiers by combining their elements into more sophisticated formulas like precision, recall and area under the ROC curve. The traditional formula for precision is:

$$P = \frac{tp}{tp + fp} \ .$$ (1)

where tp is the number of true positives and fp is the number of false positives. Precision is the ratio between the correctly predicted examples from a given class over the total number of actual examples of such class. On the other hand, recall is defined as the ratio between the number of correctly predicted examples from a given class and the total number of predicted examples for such class. Recall is often called sensitivity and is traditionally defined by:

$$TPR = \frac{tp}{tp + fn} \ .$$ (2)

where fn is the number of false negatives.

In Table 3 it is possible to observe the behavior of the algorithms with respect to precision, recall, and the area under the ROC curve. Note that all the implementation obtained relevant results.

The outstanding precision and recall values as well as the perfect area under the ROC curve demonstrate the suitability of supervised learning algorithms for the defect classification problem. In addition, it can be concluded that the set of features extracted from the original images boosts the effectiveness of the classifier.

Table 4 shows the execution time for the testing and training phases as well as the respective accuracy of the five classifiers. LIBSVM and MLP have shown excellent and similar performance with respect to the classification task, nevertheless, the efficiency of the algorithms during the testing phase is of interest as well. Note that the testing phase of AdaBoost-J48 and the SMO are by far

Table 3. Execution results for precision, recall and area under the ROC curve. The SVM parameters are shown in Table 2, Adaboost used 10 interactions and weight threshold 100 with confidence 0.25 for J48 and k=1 for IBK, MLP used 74 hidden layers, learning rate of 0.3 and momentum 0.2.

	Roc	Recall	Precision
SMO	0.9979	0.9965	0.9879
BoostIBK	0.9916	0.9876	0.9870
BoostJ48	0.9999	0.9959	0.9946
MLP	**1.0000**	0.9978	0.9978
LIBSVM	0.9991	**0.9983**	**0.9997**

the best in terms of efficiency. It is justified by the fact that the time for evaluating test examples is proportional to the number of base classifiers (decision trees) multiplied by the height of each decision tree. In the case of SVM the time for evaluating test cases is proportional to the final number of support vectors. AdaBoost-IBK presents the best time during training, not so far is the LIBSVM with the second best time. The accuracy only confirms that all the classifiers can discriminate the defects very accurately.

Table 4. Testing and training time for AdaBoosts, SVMs (The parameter selection time by SA is not included) and and MLP

	Testing time	Training time	Accuracy (%)
SMO	0.21s	2433.62s	93.10
BoostIBK	38.99s	**110.41s**	95.75
BoostJ48	**0.14s**	699.89s	98.74
MLP	1.93s	7322.86s	99.24
LIBSVM	36.70s	158.23s	**99.59**

5   Conclusion and Future Work

Previous works in solving classification problems with SVMs has shown the weakness of parameter tuning. This paper addressed this weakness and presented results for a real problem using a stochastic parameter selection for SVM training with the goal of improving the generalization performance. As expected, the use of Simulated Annealing had presented a effective solution for the problem of training SVMs.

When comparing two different implementations of SVMs, the LibSVM implementation is either the most effective algorithm or the most efficient algorithm for training purposes. Moreover, LibSVM is very suitable for the iterative process of parameter selection. Note that the difference in effectiveness between LibSVM an MLP can be neglected once both results are outstanding. One interesting observation is that AdaBoost-J48 solutions tend to be by far faster than the others classifiers for testing purposes and its loss in accuracy is very small.

In order to get faster times for evaluation a natural step in future work is the reduction of the number of features using feature selection of extraction algorithms. Clearly, efficiency is crucial for real industrial needs. Another research direction is the application of similar solutions at different leather stages which are characterized by presenting different features. An other set of experiments with a larger dataset is a must, as the low quantity of images is a problem since it does not represents the problem properly and may wrongly indicate that the problem is easy.

The DTCOURO application which already assists with image segmentation, sampling and feature extraction for the learning model generation is actually having its visual classification module finalized. Thus one will be able to apply the learned model over an input image and analyse the classification results visually over the image. The tuning module is being generalized to attend all Weka compatible classifiers as well.

References

1. Matthey, H., Fabiosa, J.F., Fuller, F.H.: Brazil: The future of modern agriculture. MATRIC (2004)
2. da Costa, A.B.: Estudo da competitividade de cadeias integradas no brasil: Impactos das zonas de livre comercio. Technical report, Instituto de Economia da Universidade Estadual de Campinas (2002)
3. Yeh, C., Perng, D.B.: Establishing a demerit count reference standard for the classification and grading of leather hides. International Journal of Advanced Manufacturing 18, 731–738 (2001)
4. Sobral, J.L.: Optimised filters for texture defect detection. In: Proc. of the IEEE International Conference on Image Processing, September 2005, vol. 3, pp. 565–573. IEEE Computer Society Press, Los Alamitos (2005)
5. Kumar, A., Pang, G.: Defect detection in textured materials using gabor filters. IEEE Transactions on Industry Applications 38(2) (2002)
6. Georgieva, L., Krastev, K., Angelov, N.: Identification of surface leather defects. In: CompSysTech 2003: Proceedings of the 4th international conference on Computer systems and technologies, pp. 303–307. ACM Press, New York (2003)
7. Krastev, K., Georgieva, L., Angelov, N.: Leather features selection for defects' recognition using fuzzy logic. In: CompSysTech 2004: Proceedings of the 5th international conference on Computer systems and technologies, pp. 1–6. ACM Press, New York (2004)

8. Branca, A., Tafuri, M., Attolico, G., Distante, A.: Automated system for detection and classification of leather defects. NDT and E International 30(1), 321–321 (1997)
9. Pistori, H., Paraguassu, W.A., Martins, P.S., Conti, M.P., Pereira, M.A., Jacinto, M.A.: Defect detection in raw hide and wet blue leather. In: CompImage (2006)
10. Osuna, E., Freund, R., Girosi, F.: Training support vector machines: an application to face detection. In: CVPR 1997, Puerto Rico, pp. 130–136 (1997)
11. Chetverikov, D.: Texture analysis using feature-based pairwise interaction maps. Pattern Recognition 32(3), 487–502 (1999)
12. Hseu, H.W.R., Bhalerao, A., Wilson, R.G.: Image matching based on the co-occurrence matrix. Technical Report CS-RR-358, Coventry, UK (1999)
13. Platt, J.: Sequential minimal optimization: A fast algorithm for training support vector machines (1998)
14. Freund, Y., Schapire, R.E.: A short introduction to boosting. Journal of Japanese Society for Artificial Intelligence 14(5), 771–780 (1999)
15. Jacinto, M.A.C., Pereira, M.A.: Industria do couro: programa de qualidade e estrat-ificacao de mercado com base em caracteristicas do couro. Simposio de producao de gado de corte, 75–92 (2004)
16. Gomes, A.: Aspectos da cadeia produtiva do couro bovino no Brasil e em Mato Grosso do Sul. In: Palestras e proposicoes: Reunioes Tecnicas sobre Couros e Peles, 25 a 27 de setembro e 29 de outubro a 1 de novembro de 2001, pp. 61–72. Embrapa Gado de Corte (2002)
17. Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P.: Optimization by simulated annealing. Science, Number 4598 220(4598), 671–680 (1983)
18. Imbault, F., Lebart, K.: A stochastic optimization approach for parameter tuning of support vector machines. In: Proceedings of the Pattern Recognition, 17th International Conference on (ICPR 2004), pp. 597–600. IEEE Computer Society Press, Los Alamitos (2004)
19. Boardman, M., Trappenberg, T.: A heuristic for free parameter optimization with support vector machines. In: Proceedings of the 2006 IEEE International Joint Conference on Neural Networks, pp. 1337–1344. IEEE Computer Society Press, Los Alamitos (2006)
20. Mitchell, T.M.: The discipline of machine learning. Technical Report CMU-ML-06-108 (2006)
21. Vapnik, V.N.: An overview of statistical learning theory. IEEE Transactions on Neural Networks 10(5), 988–999 (1999)
22. Hsu, C.W., Lin, C.J.: A comparison of methods for multi-class support vector machines. IEEE Transactions on Neural Networks 13, 415–425 (2002)
23. Witten, I.H., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques. Machine Learning 36(1-2), 105–139 (2005)
24. Amorim, W.P., Viana, R.R.R.P.H.: Desenvolvimento de um software de processamento e geracao de imagens para classificacao de couro bovino. SIBGRAPI-Workshop of Undergraduate Works (2006)
25. Witten, I.H., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques, 2nd edn. Morgan Kaufmann, San Francisco (2005)
26. Chang, C.C., Lin, C.J.: LIBSVM: a library for support vector machines, Software (2001), available at: `http://www.csie.ntu.edu.tw/~cjlin/libsvm`