Classification of Failures in Goat Leather Samples Using Computer Vision and Machine Learning

Renato F. Pereira*, Madson L. D. Dias*,
Claudio M. de Sá Medeiros*, Pedro Pedrosa Rebouças Filho*
*Programa de Pós-Graduaçao em Ciências da Computaçao,
Instituto Federal de Educação, CI6encia, Tecnologia do Ceará, Fortaleza, Ceará 60040–215
Email: {renato.francisco, madson.dias}@ppgcc.ifce.edu.br, {pedrosarf, claudiosa}@ifce.edu.br

***Abstract*—Textile industry has used goat skins in manufacturing products that require high quality control. Thus, a specialist performed a skins qualities classification to put a price on the goat leather sample, but this evaluation depends on whom evaluate. To reduce these divergences and to increase the productivity on the textile industry area, this paper presents a new approach to detect leather failure using feature extractor and machine learning classifiers. Also, a new feature extractor, called of Pixel Intensity Analyzer (PIA), is proposed for this application. Experiments were performed with a real data set comparing PIA with two other features extractors using machine learning classifiers with each one. In accuracy, the best approach was LBP with LS-SVM (RBF), but in processing time as a very important factor, since it is a real-time application to the industry, the PIA combined with ELM presents the best cost-effective because it also has excellent accuracy rates.**

I. Introduction

The production of goats is a widespread activity throughout in the whole world. In addition, the adaptability goats to climatic adversity makes this activity one of the main economic alternatives in semi-arid regions.

Goat stock in Brazil in 2007 was around 9,450,312 heads, being the world's seventh largest stock. Worldwide, goats are more concentrated in developing countries (more than 95% of stock) and tend to be more localized in dry tropical and subtropical areas of poor agricultural potential and even on marginal lands. Similarly, goats in Brazil are concentrated in the Northeast (more than 93% of stock), a region with varying agro-climatic characteristics and large semiarid areas. During the last years the number of goats has increased in all regions of Brazil between 1 and 6% [1]. Even with the high amount of leather production, Brazil still has lagged technologies in all stages of skin preparation.

The tanning is the step that consists in the transformation of the skins, previously treated, into stable and imputrescible materials, that is called leather. The leather resulting from this step is called Wet Blue and it was this kind of leather used for this work. At the end of this step, the skin is sold to the finishing industries in which the garments are produced. However, the tanning industries sell the skins with prices that vary according to the amount and localization of failures of the skins.

In most industries, leather samples are still analyzed with the naked eye by a specialists. It is a tiring process that can generate many discording between different specialists. This discording can bring financial loss to both parties, from the tannery industry that classifies and sells the sample and also to the finishing industry that may need to sample one particular class and be buying a sample from another class.

The remainder of the paper is organized as follows. The Section II presents two related works, explaining the approach of both.The Section III presents important informations about the problem. The Section IV presents an overview about the features extractor used in this work. The Section V presents an overview about the classifiers used in this work. The Section VI presents the methodology used, to classification of new samples. The Section VII presents the results achieved after the combination between some features extractor and classifiers. And finally, in the Section VIII is presented the conclusion about this work and its results obtained.

II. Related Works

It is important to say that this work functions as a follow-up or a second approach to the master's dissertation of Queiroz [2].

The qualification model proposed by Queiroz [2] works as follows: A resizing of the entire image is performed to a size of 40x40 and then the characteristics of the images are extracted using extractors like GLCM and techniques such as PCA. Then, classification is done by classifiers such as SVM and ELM.

Another important work is the of Amorin, Pistori, Pereira and Jacinto [3] that propose a classification model of goat leather failures, both in raw-hide leather images and in wet blue leather images, based in attributes on co-occurrence matrices, interaction maps, Gabor filter banks and two different color spaces. With these attributes three experiments are performed, the first one in which the raw data classification is carried out by classifiers such as KNN, C4.5, Naive Bayes and SVM, being the KNN the classifier with the highest precision rate (average of 94%). The second experiment consists of the classification of data after a reduction of attributes by methods such as FisherFace, Chen's LDA (CLDA), Direct LDA (DLDA), Yang's LDA (YLDA) and Kernel LDA (KLDA). The KNN remains the classifier that achieves higher accuracy rates (average of 91%). The third experiment compares the

location of the failures performed by the proposed model with the manual classification performed by the specialist.

III. Problem Characterization

There are numerous kind of failures that may be present in goat leather, but below it is presented the main failures, which were studied for several days with specialist follow-up for this work.

A. Main failures found in goat leather samples

The following are the main failures encountered during the problem characterization stage:

1) Wire risk: Cutting caused by wire present in fences on farms where animals live (Figure 1);

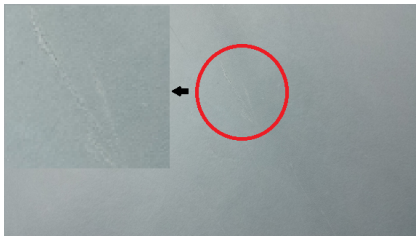

Fig. 1. Wire risk.

2) Poor conservation: Stains caused by excess salt (or low quality salt) placed in the samples for storage in the stock stage (Figure 2);

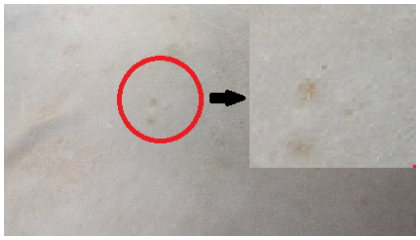

Fig. 2. Poor conservation.

3) Sign: Signs of birth on the skin of animals (Figure 3);

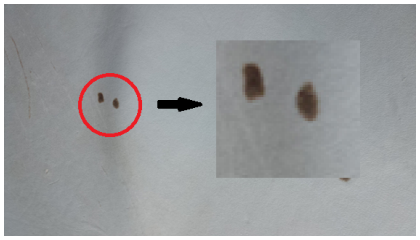

Fig. 3. Sign.

4) Bladder: Wound from skin disease (Figure 4);

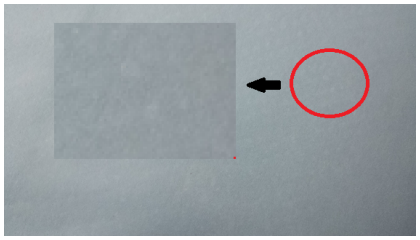

Fig. 4. Bladder.

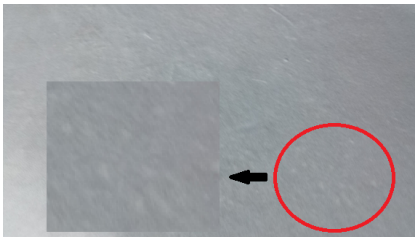

Fig. 5. Scabies.

5) Scabies: White spots caused by parasites on the animal's skin (Figure 5);

6) Mosquito bite: Micro-holes caused by mosquito bites (Figure 6);

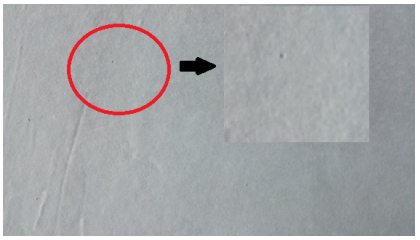

Fig. 6. Mosquito bite.

7) Scar: Wounds not fully healed (Figure 7);

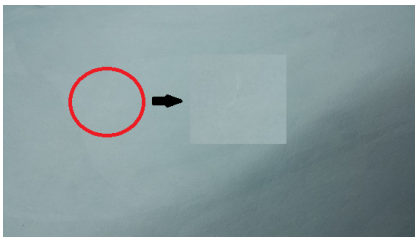

Fig. 7. Scar.

8) Rufa: Injured hair (Figure 8);

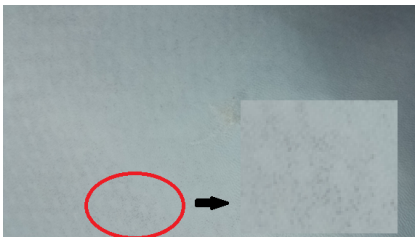

Fig. 8. Rufa.

9) Vegetable fat: Stains caused by excess vegetable fat (or low quality vegetable fat) placed in the samples for storage in the stock stage (Figure 9);

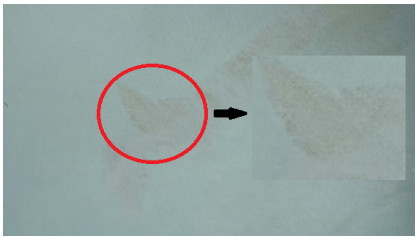

Fig. 9. Vegetable fat.

10) Hole: Hole caused by error when extracting the animal's skin.

After understanding the main failures and how the classification performed by the specialists works, the acquisition of images was done.

B. Image acquisition

The images of the goat leather samples were captured by a camera with 13MP, 4128×2322 resolution, 31mm f / 2.2 aperture, auto focus, Samsung GSIV model in a controlled environment (closed room, with always the same lights and pictures always taken in the same position).

The table (shown in Figure 10) where the samples are placed, has a bottom black, a camera holder, 1.5m away from the sample and its own lighting is done by two 40W fluorescent lamps in parallel behind the camera, so as not to generate shadows [2].

Fig. 10. Table to take pictures of the samples.

The image capture was done during a training in which the specialist characterized each failure of the sample collected. It was captured 145 images.

With every samples captured, It must be done the features extraction, presented in the next subsection.

C. Features extraction

The features extraction is the step that extracts important numeric data from image that can be using to classification.

There are many methods to features extraction and this work shows three methods, In which one of them was developed during the development this tool.

The following Figure 11 presents a diagram with the flux of the used methods.

The theoretical basis on the feature extractors is presented in Section IV.

1) Report with location of failures: The report of failure types was done as follows: The image is analyzed in its original resolution, in an image editor that informs the coordinate of the pixels observed. Failure and normal regions have their central pixel saved in the report.

This report has 4 attributes, in that the first is the identification number of the image, the second is the x position of the center of the failure, the third is the y position of the center of the failure, and finally, the fourth is the class of failure.

After this, 1874 patterns were generated, as shown in the Table

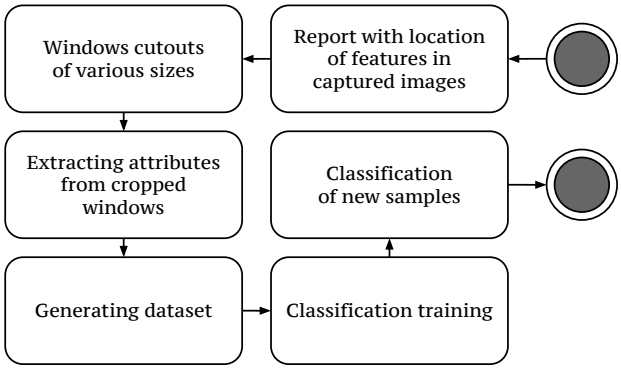

Fig. 11. Diagram with the flux of the methods used.

TABLE I
SAMPLES AMOUNT PER CLASS.

Failure	*Class*	*#Samples*
Normal	0	882
Wire Risk and Knife Cutting	1	331
Poor conservation	2	134
Sign	3	39
Bladder	4	5
Scabies	5	56
Mosquito bite	6	63
Scar	7	120
Rufa	8	84
Vegetable fat	9	5
Hole	10	155

It should be noted that the classes of Wire Risk and Knife Cut were unified by inability to distinguish in the images, although it is possible to distinguish "in loco".

It is important to say that the labels were assigned manually after 1 month of training with 2 leather classifier specialists

With the report created, an algorithm was developed to cut out the regions indicated by the report. The next subsection shows how this procedure is accomplished.

2) Windows creation with failures: In order to create windows with the failures, an algorithm has been developed to read the report and save clippings of the images with the region where the failure is.

The algorithm generates square windows in that the central pixel informed by the report is in the middle of the window and has a pre-determined size.

Six sizes of windows were tested: 51×51; 101×101; 151×151; 201×201; 251×251; 301×301. This odd number is so that the window has a central pixel, facilitating the trimming of the window.

Then it must extracting numerical characteristics of the images, for only then after this, being able to classify them. At the following topic, It showed the methods of features extraction used.

IV. OVERVIEW OF FEATURES EXTRACTION METHODS

This section presents a brief theoretical foundation about the feature extractors used in this work.

The extraction of attributes is a fundamental step because it transforms the visual characteristics of the image into numerical data, so that it can be used by the pattern classifiers.

A. Gray Level Co-Occurrence Matrix

The GLCM (Gray level Co-Occurrence Matrix) is a technique developed in the 70's by Haralick [4]. This technique analyzes the co-occurrences between pairs of pixels, that is, it does not analyze each pixel individually but rather on sets of related pixels through a distance d, known as pixel spacing (pps - pixel pair spacing), in a given direction θ that can assume the angles of 0^o, 45^o, 90^o and 135^o [5], as shown in Figure 12.

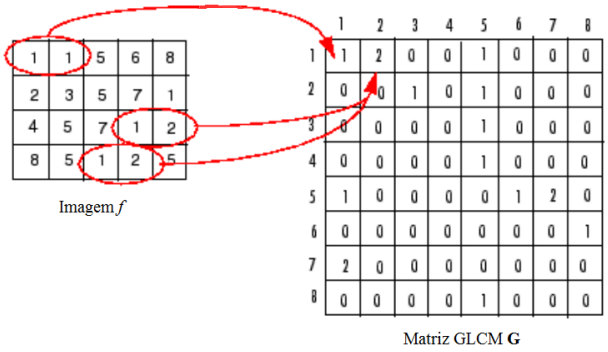

Fig. 12. Attribute Extraction Example done by GLCM.

For an input image, each cell (i, j) of the co-occurrence matrix works as a counter that stores the frequency with which two pixels appear in the image, separated by a distance d, one with the intensity value i and the other with j, as shown in Figure 12. In this illustration, on the left, there is an example image f in that appear the intensities 1 and 2 of two regions of the image, separated by one pixel of distance. Thus, if $d = 1$ representing the pixel immediately to the right of the co-occurrence G, has its position $(1, 2)$ increased to 2, indicating the occurrence of 2 pairs with color 1 and 2 separated by distance d. This process is repeated until the matrix of co-occurrence G be complete [6].

One of the elements of the GLCM matrix is the number of transitions between the gray levels, however, the texture characteristics are obtained from another representation of the matrix G, determined by its normalized matrix. The calculation for this is the number of times this co-occurrence happens divided by the total number of possible combinations, as shown in Equation 1 below.

$$P_{ij} = \frac{P(i,j)}{\sum_{i=0}^{L}\sum_{j=0}^{L} P(i,j)}, \tag{1}$$

where L represents the maximum gray level present in the image. It is also possible to relationship between three or more pixels, but it is not very efficient therefore, this approach is not used in practice.

The GLCM is always a square matrix with $L \times L$ elements, in that L is the number of levels of intensity. For example, for an image stored in 256 tones of gray, the GLCM has dimension 256 x 256. This brings a great disadvantage to GLCM: The big space needed for your storage. In addition, an image can generate a sparse array that is a matrix with many values equal to zero. Another problem in GLCM is the processing time required for its calculation. In order to decrease the computational load, a methodology widely used is to quantify the intensities in some bands, aiming to control the dimensions of the matrix G.

In total, 14 statistical measures were proposed by Haralick [4] to the GLCM. These characteristics are significant and the amount used in a problem change according to your specifications.

B. Local Binary Patterns

LBP (Local Binary Patterns) are simple and efficient texture descriptors. Formulated by Ojala [7], the LBP attributes are labeled (represented by a binary number) to each analyzed pixel. To assign this label, it is checked if each neighbor of the pixel in analysis, in a radius considered, exceeds or not, a threshold that is determined by the value of the central pixel whose label is being determined. Each neighbor that exceeds or equals this threshold is marked with the binary digit 1, otherwise it gets 0. The junction of these binary digits represents the pixel central. It is necessary to follow a logic to determine which digit is the most significant, in other words, from which neighbor the number begins to be formed, since the neighborhood is circular. It should also be noted that when determining the most automatically, a weight is assigned to each of the digits. The sum of the multiplication of each digit by its respective weight returns what is referenced as LBP code. Since the neighborhood consists of 8 pixels, a total of 2^8 = 256 patterns can be encoded. The process is shown in Figure 13.

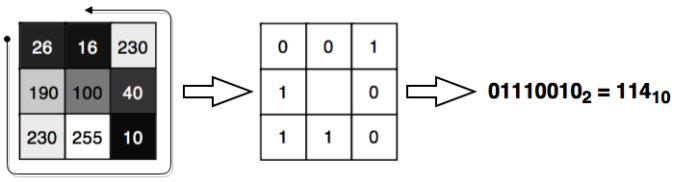

Fig. 13. Attribute Extraction Example made by the 3x3 neighborhood LBP.

Considering an image in gray levels **I**, let g and c be the intensity value of any pixel ($g_c = i(x, y)$), g_p for p = 0,..., 7 denotes the intensity value in the space neighborhood $3x3$ of (x, y). The LBP label of the neighborhood intensity pattern space of the pixel (x, y) is given by (Equation 2)

$$LBP = \sum_{p=0}^{P-1} s(g_p - g_c)2^p, \tag{2}$$

in that $P = 8$ and $s(\cdot)$ is a degree function (Equation 3)

$$s(x) = \begin{cases} 1, & \text{if } x \geq 0 \\ 0, & \text{if } x < 0 \end{cases} \tag{3}$$

The LBP operator can be easily generalized to any circular neighborhood P with radius R. The standard g_p is then defined as (Equation 4):

$$g_p = \mathbf{I}(x_p, y_p) \; for \; p = 0, \ldots, P-1 \tag{4}$$

in that (Equation 5 and 6)

$$x_p = x + R\cos\left(\frac{2\pi p}{P}\right) \tag{5}$$

$$y_p = y + R\sin\left(\frac{2\pi p}{P}\right) \tag{6}$$

The values of g_p are interpolated. The Equation 2 is used to compute the LBP.

C. Pixel Intensity Analyzer

PIA (Pixel Intensity Analyzer), is a simple extractor which gives a good precision for the classification of patterns and has a good speed of features extraction for this application. Developed to this work, PIA works the following way - It is applied the Sobel Operator followed by Canny Detector [8] to find edges in the images. After of the application of these techniques, the edges found are painted with different colors, according with your positions, intensities and connections, for example, a small threshold for edge location is used, and through an iterative process it increases. As it increases, the edges begin to appear according to the intensity. The first edge that appears is painted one color, the second one another, and so on., according with the Figures 14, 15, 16 and 17.

Fig. 14. Preprocessing done by PIA extractor - Normal sample.

Fig. 15. Preprocessing done by PIA extractor - Normal sample with noise.

Fig. 16. Preprocessing done by PIA extractor - Sample with Failure (Scar).

Fig. 17. Preprocessing done by PIA extractor - Sample with Failure (Hole).

So, after apply this method, is calculated the summation of all values of channel 1 of the image(Blue), how is showed in the Equation 7

$$\mathrm{A}_1 = \sum_{i=1}^{N}\sum_{j=1}^{M} P_{B_{ij}}, \tag{7}$$

the summation of all values of channel 2 of the image(Green), how is showed in Equation 8

$$\mathrm{A}_2 = \sum_{i=1}^{N}\sum_{j=1}^{M} P_{G_{ij}}, \tag{8}$$

and the summation of all values of channel 3 of the image(Red), how is showed in Equation 9

$$\mathrm{A}_3 = \sum_{i=1}^{N}\sum_{j=1}^{M} P_{R_{ij}}. \tag{9}$$

And in following, is computed the max value per channel, how is showed in the Equations 10, 11 and 12.

$$A_4 = \max\left\{P_{B_{ij}}\right\}, \;\; i = 1, \ldots, M \;\; \text{and} \;\; j = 1, \ldots, N, \tag{10}$$

$$A_5 = \max\left\{P_{G_{ij}}\right\}, \;\; i = 1, \ldots, M \;\; \text{and} \;\; j = 1, \ldots, N, \tag{11}$$

$$A_6 = \max\left\{P_{R_{ij}}\right\}, \;\; i = 1, \ldots, M \;\; \text{and} \;\; j = 1, \ldots, N. \tag{12}$$

in that N is the rows number of the image, M is the columns number of the image, P is the analyzed pixel, B is the representation of blue channel, G is the representation of green channel and R is the representation of red channel of the image.

And this way works the PIA, that works well because in images of normal class, normally don't have much edges when compared with the images of the failure class and to extract these information of a small image is a very fast task.

The next Section presents an overview about the classifiers used.

V. Pattern Classification Methods Applied

This section presents a brief theoretical foundation about the classifiers used in this work.

A. k-Nearest Neighbors

The first classifier used in our simulation is the k-nearest neighbors (KNN, [9]). The KNN is a non-parametric method used for classification and regression tasks. The main idea of KNN algorithm is storage the complete training data and classify new examples by choosing the majority class among the k closest examples in the stored data set. Normally, the Euclidean distance is used as similarity metric.

B. Minimal Learning Machine

Minimal learning machine (MLM, [10]) is a recent approach of a supervised learning algorithm, that can be applied to classification and regression problems. The basic operation of MLM consists in the presupposition of the existence of a mapping between the geometric configurations of points in input space and the geometric configurations of respective points in output space. This mapping is represented by two distance matrix (input and output), computed for all points in the training set and a subset of it, named *reference points*.

The learning of MLM is accomplished by determination of a regression linear model between of two distance matrix. Thus, given a point in the input space, the MLM can calculate the location of this point in the output space, through the learned regression model. In other words, the MLM learning algorithm consists in obtaining a multi-response linear system solution. The only hyperparameter of MLM is the number of reference points K. MLM can be extended for classification problems representing the S classes in vector form through the codification binary scheme (1-of-S).

C. Radial basis function

A network of radial basis function (RBF, [11]) is a hybrid neural architecture that combines concepts of supervised and unsupervised learning. This method is widely used for function approximation and pattern recognition. In the original proposal, the RBF presents a hidden layer of non-linear activation functions and a output layer of linear functions (in general).

The parameters of this method are the number of RBF are the hidden neuron number, the type of activation function and the form of of the radial basis function centroids are selected.

D. Extreme Learning Machines

Extreme Learning Machines (ELM, [12], [13]) are single-hidden layer feedforward networks that have universal approximation capability based on the ordinal least squares. Its architecture relies on random and fixed-weight hidden neurons and non-linear activation functions. ELMs have two main advantages. The first one is the training speed, which is much faster than most learning algorithms such as MLPs, Support Vector Machines [14] and so on. The second advantage is the absence of parameters to be tuned, but the number of hidden neurons.

E. Support Vector Machines

Support vector machines (SVM, [14], [15]) are based on the *Structural Risk Minimization* principle from statistical learning theory. According to Vapnik [14], the risk of a learning machine (R) is bounded by the sum of the empirical risk estimated from training samples (R_{emp}) and a confidence interval $(\Psi) : R \leq R_{\text{emp}} + \Psi$. The strategy of SRM is to keep the empirical risk (R_{emp}) fixed and to minimize the confidence interval (Ψ), or to maximize the margin between a separating hyperplane and closest data points.

In their basic formulation, SVMs learn linear threshold function. Nevertheless, by a simple application of an appropriate kernel function K, they can be used to learn polynomial classifiers, radial basic function (RBF) networks, and three-layer sigmoid neural nets. An example of such a kernel function is $K(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\gamma \|\mathbf{x}_i - \mathbf{x}_j\|^2\right)$.

F. Least Squares Support Vector Machines

Similarly to SVM, the Least Squares Support Vector Machine (LSSVM [16]) classifiers are also able to solve classification problems and function approximations. LSSVM is a viable alternative for the standard SVM formulation, due to a modification of the Karush-Kuhn-Tucker (KKT) restrictions. In this way, the he training phase is summarized by a solution system of linear equations. In contrast, the quadratic programing used in SVM has a high computational cost.

It is worth noting that the classifiers used are based on different paradigms, which is important to understand the behavior of each one in front of this problem.

VI. Classification of new samples

To classify a new sample, it is necessary a series of image preprocessing before the classification. In the Figure 18 is presented a diagram with the flux of image preprocessing.

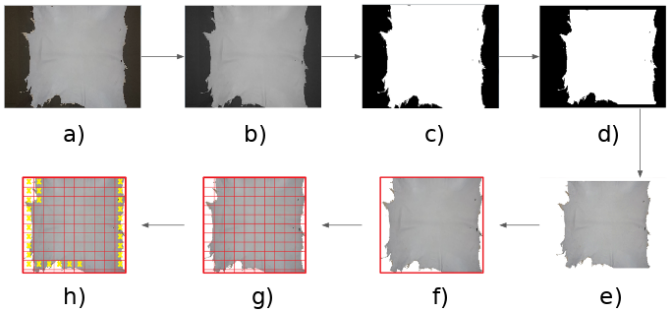

Fig. 18. Diagram with the flux of image preprocessing.

The Figure 18 (a) is the original image, that in following is passed to gray scale (Figure 18 (b)) and used a Otsu Threshold [17] (Figure 18 (c)), which is an adaptive thresholding that choose the optimal value automatically for the threshold, taking the image in black and white.

After the thresholding, the leather sample stay in white and the background in black. A 5×5 mean mask filter is also used

to remove noises that may appear in the black region (just the enough to remove noise of captured images).

It is noticed that in some cases the leather sample touches the ends of the window. To avoid having problems with edge detection, 10 lines of pixels are added at the bottom and top of the image, as well as 10 columns of pixels are added at left and right sides of the image. The new image is shown in Figure 18(d).

With the object well defined, the image is brought back to 3 channels (colored) through of an iterative process pixel-by-pixel, with the following rule: Where it is white, the new image receives the pixel of the original image and where it is black, the new image receives black. To avoid confusing the external background with holes in leather, is applied the technique of Region Growing [18] in the seed pixel (0, 0) (seed pixel is the one which initiates the growing process), so what it is outside of the sample stay white. This technique consists of a iterative process in that the pixel color of the seed is checked and the color change is made to the desired color while the iteration verifies that the neighboring pixels have the same color of the seed. In this way, it is seen that the image is as shown in Figure 18 (e).

Using a convex shell technique, a rectangle is drawn around the leather. In Figure 18 (f) the image with the rectangle is shown the entire convex area of the sample.

The methodology adopted involves dividing the sample into smaller rectangles, one by one, extracting their attributes and classifying them. In Figure 18 (g), the image with a 10x10 grid is shown as an example for better viewing effect. In the practice, the grid will be of the same size of the windows used to training.

The next step is to analyze these rectangles. The rectangle having at least one white pixel, is discarded. Therefore, it is only considered the rectangle that has 100% of the image filled by leather. Then the rectangle that has some white pixel have a yellow X drawn, showing that it is not used for classification of the sample, as shown in Figure 18 (h).

Once this is done, must extract the attributes of each rectangle and after, classify, this way is possible to know the amount of failures and their positions.

VII. Results

All results presented are for 51×51 windows, because with this window size, it has been better results. Were used 9 attributes of the GLCM, 48 attributes of the LBP and 9 attributes of the PIA.

In our simulations, 80% of the data examples were randomly selected for training purposes and the remaining 20% of the examples were used for assessing the classifiers' generalization performances. We carried out 30 executions on each data set. The classifier parameters were tuned by applying grid search with 5-fold cross-validation over the training dataset.

Table II compares the performance of PIA against GLCM and LBP feature extractors. The classifier parameters were tuned by applying grid search with 5-fold cross-validation over the training dataset. For KNN, the neigbohood number is searched in $[1, 3, \ldots, 15]$ range and the Euclidean norm is adopted to calculate the distance. For MLM the number of reference points are searched in a grid formed by $[10\%, 20\%, \ldots, 100\%]$ of the training dataset size. For RBF, the gaussian function are used, the centers are randomly chosen and the number of centers are searched in a grid formed by $[2, 4, \ldots, 300]$. For ELM the sigmoid activation function are adopted and the hidden neuron number are searched in a grid formed by $[2, 4, \ldots, 300]$. For SVM and LSSVM with linear kernel parameter of regularization are searched in a grid formed by $[2^{-5}, 2^{-3}, \ldots, 2^{15}]$. Finaly, for SVM and LSSVM with with gaussian kernel, the regularization parameter also searched in $[2^{-5}, 2^{-3}, \ldots, 2^{15}]$ and the kernel aperture (γ) is searched in a grid $[2^{-15}, 2^{-3}, \ldots, 2^{3}]$.

Considering that TP is the *True Positive*, TN is the *True Negative*, FP is the *False Positive* and FN is the *False Negative*, the metrics can be calculated as follows:

$$\text{ACC} = \frac{TP + TN}{TP + TN + FP + FN}, \tag{13}$$

$$\text{PRE} = \frac{TP}{TP + FP}, \tag{14}$$

$$\text{SPE} = \frac{TN}{TN + FP}, \tag{15}$$

$$F_1 = \frac{2 \cdot TP}{2 \cdot TP + FP + FN}. \tag{16}$$

Other important results are the times per classifier and features extractor, because as said in subsection of Image Acquisition, the images has size of 4128×2322, that is, many windows fit inside of an image of these. So, for do not delay the production (the specialist takes around between 20 and 60 seconds to analyze each image) the classification must be fast. In the Table III are found the times per classifier and in the Table IV is shown the average times to features extraction of an image 51×51 at the validation moment.

VIII. Conclusion

According to the tables presented in the previous section, the best results are the combination of the LBP extractor and the LS-SVM classifier with the RBF kernel. This combination has better accuracy rates, however, looking the time cost, It is a few high, already the combination of the PIA extractor and the ELM classifier has a good accuracy rates, near of the previous combination and lower execution time. Then one can affirm that the choice stay at the criteria of the expert, whether he prefers to sacrifice a little time for better accuracy rates or to sacrifice a little accuracy because of shorter classification times.

With the failures classified according with your location, it is possible aid the specialist in the quality classification, avoiding differences with other experts and with the finishing industry.

TABLE II
ACCURACY (ACC), PRECISION (PRE), SPECIFICITY (SPE) AND F-SCORE (F_1) IN PERCENTAGE WITH YOUR VARIANCES RESPECTIVELY FOR KNN, MLM, RBF, ELM, SVM AND LSSVM WITH LINEAR KERNE (SVM^{lin} AND LSSVM^{lin}, RESPECTLY), SVM AND LSSVM WITH RBF KERNEL (SVM^{rbf} AND LSSVM^{rbf}, RESPECTLY).

Classifier	*Metric*	PIA	GLCM	LBP
k-NN	ACC	84.56 ± 1.58	67.20 ± 2.12	88.27 ± 1.16
	PRE	86.48 ± 2.41	69.32 ± 4.40	91.41 ± 1.83
	SPE	82.93 ± 2.98	65.36 ± 3.83	85.55 ± 2.03
	HAM	84.61 ± 1.57	67.10 ± 2.18	88.36 ± 1.14
MLM	ACC	84.86 ± 1.70	65.76 ± 2.76	89.06 ± 1.44
	PRE	85.20 ± 2.32	65.31 ± 5.20	89.25 ± 2.63
	SPE	84.60 ± 3.08	66.23 ± 3.36	88.92 ± 1.99
	HAM	84.85 ± 1.69	65.60 ± 2.85	89.05 ± 1.47
RBF	ACC	85.23 ± 1.37	66.70 ± 2.34	88.80 ± 1.14
	PRE	89.01 ± 2.28	67.76 ± 4.28	89.92 ± 2.08
	SPE	81.96 ± 2.84	65.78 ± 4.20	87.85 ± 2.06
	HAM	85.28 ± 1.38	66.57 ± 2.27	88.84 ± 1.13
ELM	ACC	84.03 ± 1.66	73.97 ± 1.87	86.99 ± 1.48
	PRE	87.50 ± 2.32	75.56 ± 4.16	92.46 ± 2.07
	SPE	81.04 ± 3.34	72.63 ± 3.00	82.21 ± 2.42
	HAM	84.08 ± 1.69	73.94 ± 1.88	87.00 ± 1.37
SVM^{lin}	ACC	80.61 ± 1.83	68.36 ± 1.93	86.13 ± 1.45
	PRE	94.55 ± 1.82	73.17 ± 3.24	92.85 ± 1.98
	SPE	68.36 ± 3.80	64.13 ± 3.35	80.26 ± 2.36
	HAM	79.26 ± 2.27	68.25 ± 1.96	86.06 ± 1.41
SVM^{rbf}	ACC	83.82 ± 1.55	67.95 ± 2.21	89.06 ± 1.23
	PRE	88.34 ± 2.75	74.19 ± 3.38	90.89 ± 2.25
	SPE	79.90 ± 3.12	62.46 ± 3.76	87.47 ± 2.39
	HAM	83.84 ± 1.58	67.71 ± 2.34	89.10 ± 1.21
LSSVM^{lin}	ACC	76.84 ± 2.26	67.98 ± 1.88	84.66 ± 1.53
	PRE	96.16 ± 1.27	62.15 ± 4.09	94.13 ± 1.95
	SPE	59.93 ± 3.63	73.26 ± 3.04	76.35 ± 2.69
	HAM	73.77 ± 2.61	67.11 ± 2.18	84.27 ± 1.56
LSSVM^{rbf}	ACC	85.61 ± 1.39	68.20 ± 2.05	89.32 ± 1.18
	PRE	84.70 ± 2.47	70.15 ± 3.92	91.23 ± 2.36
	SPE	86.38 ± 2.58	66.50 ± 3.97	87.71 ± 2.42
	HAM	85.48 ± 1.42	68.12 ± 2.11	89.39 ± 1.13

TABLE III
TESTING TIMES IN MILLISECONDS WITH YOUR VARIANCES RESPECTIVELY FOR KNN, MLM, RBF, ELM, SVM AND LSSVM WITH LINEAR KERNE (SVM^{lin} AND LSSVM^{lin}, RESPECTLY), SVM AND LSSVM WITH RBF KERNEL (SVM^{rbf} AND LSSVM^{rbf}, RESPECTLY).

Classifier	PIA	GLCM	LBP
k-NN	30.48 ± 2.23	52.00 ± 15.48	93.70 ± 32.34
MLM	3.87 ± 2.22	4.49 ± 3.38	32.17 ± 21.51
RBF	1.02 ± 1.08	2.04 ± 1.44	1.16 ± 0.55
ELM	0.65 ± 0.27	1.69 ± 1.08	2.96 ± 2.05
SVM^{lin}	3.87 ± 1.93	3.48 ± 0.31	6.25 ± 2.06
SVM^{rbf}	22.38 ± 5.50	18.89 ± 2.88	28.53 ± 7.61
LSSVM^{lin}	3.14 ± 0.41	3.22 ± 0.35	4.11 ± 1.37
LSSVM^{rbf}	8.51 ± 2.55	13.53 ± 4.42	11.88 ± 1.66

TABLE IV
TIMES IN MILLISECONDS TO FEATURES EXTRACTION PER SAMPLE.

Features extractor	*Times*
GLCM	22.33 ± 2.33
LBP	1.87 ± 0.63
PIA	0.03 ± 0.01

ACKNOWLEDGMENT

The authors would like to thank industry CV Couros for providing the training to identify the failures and the environment for image capture.

REFERENCES

[1] R. B. Lôbo, O. Facó, A. B. O. Lôbo, and L. V. Villela, "Brazilian goat breeding programs," *Small Ruminant Research*, vol. 89, no. 2, pp. 149–154, 2010.

[2] *A Vision Computer System to Quality Classification of Goat Leather.* Fortaleza, CE, Brazil: UFC, August 2013.

[3] W. P. Amorim, H. Pistori, M. C. Pereira, and M. A. C. Jacinto, "Attributes reduction applied to leather defects classification," in *Graphics, Patterns and Images (SIBGRAPI), 2010 23rd SIBGRAPI Conference on*. IEEE, 2010, pp. 353–359.

[4] R. M. Haralick, K. Shanmugam, and I. H. Dinstein, "Textural features for image classification," *Systems, Man and Cybernetics, IEEE Transactions on*, no. 6, pp. 610–621, 1973.

[5] A. Baraldi and F. Parmiggiani, "An investigation of the textural characteristics associated with gray level cooccurrence matrix statistical parameters," *Geoscience and Remote Sensing, IEEE Transactions on*, vol. 33, no. 2, pp. 293–304, 1995.

[6] L. Billy, G. Arvisenet, P. Poinot, S. Chevallier, G. Royer, E. Vigneau, and C. Prost, "Image texture analysis of apples broken down in a mastication simulator prototype," in *13th World Congress of Food Science & Technology 2006*, 2006, pp. 1358–1358.

[7] T. Ojala, M. Pietikainen, and D. Harwood, "Performance evaluation of texture measures with classification based on kullback discrimination of distributions," in *Pattern Recognition, 1994. Vol. 1 - Conference A: Computer Vision amp; Image Processing., Proceedings of the 12th IAPR International Conference on*, vol. 1. Jerusalem, Israel: IEEE, Oct 1994, pp. 582–585 vol.1.

[8] J. Canny, "A computational approach to edge detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 8, no. 6, pp. 679–698, Jun. 1986. [Online]. Available: http://dx.doi.org/10.1109/TPAMI.1986.4767851

[9] T. M. Cover and P. E. Hart, "Nearest neighbor pattern classification," *IEEE Trans. Information Theory*, vol. 13, no. 1, pp. 21–27, 1967. [Online]. Available: https://doi.org/10.1109/TIT.1967.1053964

[10] A. H. de Souza Junior, F. Corona, G. D. A. Barreto, Y. Miché, and A. Lendasse, "Minimal learning machine: A novel supervised distance-based approach for regression and classification," *Neurocomputing*, vol. 164, pp. 34–44, 2015.

[11] J. Park and I. W. Sandberg, "Universal approximation using radial-basis-function networks," *Neural Computation*, vol. 3, no. 2, pp. 246–257, 1991. [Online]. Available: https://doi.org/10.1162/neco.1991.3.2.246

[12] G. Huang, Q. Zhu, and C. K. Siew, "Extreme learning machine: Theory and applications," *Neurocomputing*, vol. 70, no. 1-3, pp. 489–501, 2006. [Online]. Available: https://doi.org/10.1016/j.neucom.2005.12.126

[13] G. Huang and L. Chen, "Convex incremental extreme learning machine," *Neurocomputing*, vol. 70, no. 16-18, pp. 3056–3062, 2007. [Online]. Available: https://doi.org/10.1016/j.neucom.2007.02.009

[14] V. Vapnik, "The nature of statistical learning theory," *Data mining and knowledge discovery*, 1995.

[15] B. E. Boser, I. Guyon, and V. Vapnik, "A training algorithm for optimal margin classifiers," in *Proceedings of the Fifth Annual ACM Conference on Computational Learning Theory, COLT 1992, Pittsburgh, PA, USA, July 27-29, 1992.*, 1992, pp. 144–152. [Online]. Available: http://doi.acm.org/10.1145/130385.130401

[16] J. A. K. Suykens and J. Vandewalle, "Least squares support vector machine classifiers," *Neural Processing Letters*, vol. 9, no. 3, pp. 293–300, 1999. [Online]. Available: https://doi.org/10.1023/A:1018628609742

[17] D. Liu and J. Yu, "Otsu method and k-means," in *Proceedings of the 2009 Ninth International Conference on Hybrid Intelligent Systems - Volume 01*, ser. HIS '09. Washington, DC, USA: IEEE Computer Society, 2009, pp. 344–349. [Online]. Available: http://dx.doi.org/10.1109/HIS.2009.74

[18] R. Adams and L. Bischof, "Seeded region growing," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 16, no. 6, pp. 641–647, 1994.