2017 Workshop of Computer Vision

Biometric Recognition based on Fingerprint: A Comparative Study

Bruno Matarazzo Durú
University of São Paulo - EACH
São Paulo, Brazil
Email: brduru@bol.com.br

Jonas Mendonça Targino
University of São Paulo - EACH
São Paulo, Brazil
Email: jonas.mendonca@usp.br

Clodoaldo Aparecido de Moraes Lima
University of São Paulo - EACH
São Paulo, Brazil
Email: c.lima@usp.br

***Abstract*—Fingerprint recognition is regarded as one of the most popular and reliable techniques for automatic personal identification due to the well-known distinctiveness and persistence of fingerprints. A critical issue in biometric system design is the choice of classifier. In this paper, we conducted a systematic performance evaluation of the five classifiers (Neural Networks, Support Vector Machines configured with radial basis function, Optimum Path Forest, K-nearest neighbors and Extreme Learning Machine) for the task of biometric recognition based on fingerprint. Experimental results conducted on a publicly available database are reported whereby we observe that the Support Vector Machine significantly outperform the others classifiers according to accuracy measure calculated.**

I. Introduction

Biometric recognition refers to the automated identification of individuals based on their physical and behavioral characteristics such as fingerprint, face, iris, and voice. Biometric recognition offers an alternative to traditional methods, such as passports, ID cards, driving licenses or PIN numbers. Thus, biometric recognition can be used to identify individuals in surveillance operations where covert recognition is required or in scenarios where a person may attempt to conceal their true identity (e.g., by using forged documents to claim social welfare benefits). Consequently, the application domain of biometrics far exceeds that of passwords and tokens. Fingerprint recognition is regarded as one of the most popular and reliable techniques for automatic personal identification due to the well-known distinctiveness and persistence of fingerprints.

A biometric system comprises of (i) Image acquisition module: this acquires the image of a biometric modality and submits it to the system for further processing,(ii) Feature extraction module: processes the acquired image thereby extracting the salient or discriminatory features,(iii) Matcher module: matches the extracted features of probe image with those of gallery image to obtain a match score whereas,an embedded decision making module verifies or rejects the claimed identity based on the match score and(iv) Database module: contains the digital representation of previously acquired samples very often termed as templates.

In order to obtain higher accuracy, various features, such as the FingerCode, ridge distributions and directional images, have also been actively investigated. Jain et al. [12] proposed the FingerCode; a method that uses a Gabor filter to extract directional ridge flow, and Park [21] used an orientation filtered by a fast Fourier transform. Chong et al. [6] employed both a geometric grouping and a global geometric shape analysis of fingerprint ridges, while Cappelli et al. [2] proposed a directional image that models fingerprints with a graph. Nagaty [16] extracted a string of symbols using block directional images of fingerprints, while Chang and Fan [3] proposed a ridge distribution model consisting of a combination of 10 basic ridge patterns with different ridge distribution sequences. In this paper, we use the approach proposed by Jain et al. [12] for feature extraction.

One of the problems presented while recognizing a fingerprint it is a recognition of patterns, since the purpose of this is to classify objects Of interest in one of several categories or classes. Generally, the objects of interest are called patterns and in the case of fingerprint recognition are called vector codes or Fingercodes that are extracted from an input image of fingerprint using feature extraction techniques.

The classification problem can be categorized as binary classification problems (two-class classification) and multiclass classification problems. Nowadays, binary or muticlass classification problem can be solved by many algorithms. For instance, Neural networks, Support Vector Machines, Optimum Path Forest, K-nearest neighbors and Extreme Learning Machine. A critical issue in biometric system design is the choice of classifier.

In this paper, we focus mostly on the classification stage, taking into account the highly complex behavior displayed by fingerprint. The approach consists of two main modules: a feature extractor based on FingerCodes that generate a feature vector from an image of fingerprint and a classifier that produces the class based on the features vector.

In the Section II, we describe the feature extraction technique based on FingerCodes. In Section III we present the classifiers; in Section IV we show the results of the classifications and the accuracy rate; and then in Section V we conclude the study.

II. Feature Extraction

It is desirable to obtain representations for fingerprints which are scale, translation, and rotation invariant. Scale invariance is not a significant problem since most images of fingerprint could be scaled as per the dpi specification of the sensors. The rotation and translation in

0-7695-6357-0/17/$31.00 ©2017 IEEE
DOI 10.1109/WVC.2017.00022

be accomplished by establishing a reference frame based on the intrinsic fingerprint characteristics which are rotation and translation invariant. It is also possible to establish many frames of reference based upon several landmark structures in a fingerprint to obtain multiple representations.

The four main steps in FingerCode [12] feature extraction algorithm are:

1) Determine a reference point and region of interest for the fingerprint image;
2) Tessellate the region of interest around the reference point;
3) Filter the region of interest in eight different directions using a bank of Gabor filters (eight directions are required to completely capture the local ridge characteristics in a fingerprint while only four directions are required to capture the global configuration);
4) Compute the average absolute deviation from the mean of grey values in individual sectors in filtered images to define the feature vector or the FingerCode.

III. Classification Algorithms

In this section we revise the use of K-nearest neighbors, Optimum Path Forest, Neural Networks, Support Vector Machine and Extreme Learning Machine in classification problems.

A. K-nearest neighbors - k-NN

k-NN was formally proposed more than 60 years ago and is still a very popular and a very studied classifier. The literature presents many applications using k-NN, such as breast cancer diagnosis [22], text classification [26] and [9], emotion recognition [4], speaker identification [13], among many others. One of the main weaknesses of the k-NN classifier is that all the training samples have to be stored in memory, and to perform classification it is necessary the computation of the distance of the test sample to all training samples. Then, look for the k samples that are closer, and finally perform a voting scheme to decide the class of the test sample. As the number of samples in the training set increases, storing all its values in the computer memory may not be feasible and also the classification procedure may take too much time due to the distances computation.

B. Optimum Path Forest - OPF

Unlike k-NN, the Optimum Path Forest (OPF) is a very recent classifier proposed in the 2000s by [18]. It is non-parametric, fast, simple, multi-class, does not make any assumption about the shapes of the classes, and can handle some degree of overlapping between classes [18]. OPF has been successfully used in many applications, such as laryngeal pathology detection [20], face recognition [17], rainfall estimation [8], image categorization [19], among many others.

The OPF is a classification technique based on graphs, being responsible for reducing, in order to classify each class, that is, the prototype according to one or more trees. So each node of the graph represents a sample of the training set acting more specifically as the descriptor, belonging to the tree of the prototype in that it is higher degree of connection according to the edges that are the distances between the descriptors.

C. Neural Networks

A neural network is a computational structure inspired by the study of biological neural processing. There are many different types of neural networks, from relatively simple to very complex, just as there are many theories on how biological neural processing works. A layered feed-forward neural network has layers, or subgroups of processing elements. A layer of processing elements makes independent computations on data that it receives and passes the results to another layer. The next layer may in turn make its independent computations and pass on the results to yet another layer. Finally, a subgroup of one or more processing elements determines the output from the network. Each processing element makes its computation based upon a weighted sum of its inputs. The first layer is the input layer and the last the output layer. The layers that are placed between the first and the last layers are the hidden layers. The processing elements are seen as units that are similar to the neurons in a human brain, and hence, they are referred to as cells, or artificial neurons. A threshold function is sometimes used to qualify the output of a neuron in the output layer. Synapses between neurons are referred to as connections, which are represented by edges of a directed graph in which the nodes are the artificial neurons. Neural networks consist of small units called neurons, and these are connected to each other in such a way that they can pass signals to each other [1].

A feed forward single layer perceptron trained with Back-propagation and Levenberg-Marquardt algorithm [10], was used in this work. The Back-propagation algorithm used in the training of multilayer perceptron, is formulated as a nonlinear least squares problem. Essentially, the Levenberg-Marquardt algoritm is a least-squares estimation method based on the maximum neighborhood idea. Let $E(w)$ be an objective error function made up of m individual error terms $e_i^2(w)$ as follows:

$$E(w) = \sum_{i=1}^{m} e_i^2(w) = ||f(w)||^2$$

where

$$e_i^2(w) = (y_i - y_{ri})^2$$

y_i is desired value of output neuron i and y_{ri} is the actual output of that neuron. It is assumed that function $f(.)$ and its Jacobian J are known at point w. The aim of the Levenberg-Marquardt algorithm is to compute the weight vector w such that $E(w)$ is minimal. In each iteration the weight vector is updated according to (1).

$$w_{k+1} = w_k + \delta w_k \tag{1}$$

where

$$\delta w_k = -(J_k^T f(w_k))(J_k^T J_k + \lambda I)^{-1} \tag{2}$$

J_k is the Jacobian of $f(.)$ evaluated at w_k, λ is the Marquardt parameter, and I is the identity matrix.

D. Support Vector Machine

Given a training data set composed of N samples $\{x_i, y_i\}_{i=1}^{N}$ with input $x_i \in \mathcal{R}^n$ and output $y_i \in \pm 1$, the SVM classifier aims at constructing a decision surface of the form $sign[f(x; w)]$, where $f(x; w) = w^T \phi(x) + b$ is an approximation to the mapping function y, $w \in \mathcal{R}^m$, and $\phi(.) : \mathcal{R}^n \rightarrow \mathcal{R}^m$ is a function mapping the input into a called higher dimensional feature space. The parameter w and b can be obtained through the following optimization problem [24]:

$$\min_{w,b,\xi} \Phi(w, b, \xi) = \frac{1}{2}(w^T w) + C \sum_{i=1}^{N} \xi_i, \qquad (3)$$

subject to $y_i[w^T \phi(x_i) + b] \geq 1 - \xi_i, \xi_i \geq 0, i = 1, \cdots, N$, where C is a trade-off parameter indicating the relative importance of the model's complexity when compared to the training error, and ξ_i is the training error for the i-th sample. For simplicity, the problem (3) is usually converted into an equivalent problem defined in a dual space, by constructing the following Lagrangian [24]: $L(w, b, \xi, \beta, \gamma)$

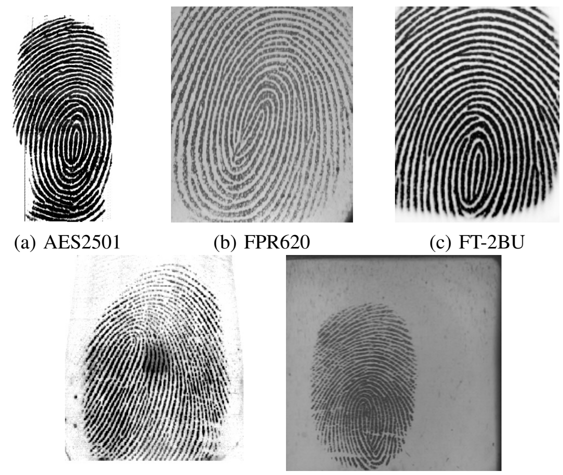
$$\frac{1}{2} w^T w + C \sum_{i=1}^{N} \xi_i - \sum_{i=1}^{N} \beta_i \{[w^T \phi(x_i) + b] y_i - 1 + \xi_i\} - \sum_{i=1}^{N} \gamma_i \xi_i \qquad (4)$$

where $\beta_i \geq 0$, $\gamma_i \geq 0$, $(i = 1, \cdots, N)$ are Lagrange multipliers. In such information, a particular kind of function, known as kernel, is employed [23]. It should follow the constraint imposed by Mercer's Theorem and provides a one-step implicit calculation of the product between $\phi(x_i)$ and $\phi(x_j)$: $K(x_i, x_j) =< \phi(x_i), \phi(x_j) >$. The results discussed in Section IV were obtained with the RBF Kernel: $K(x_i, x_j) = exp\left(-\frac{(x_i - x_j)^T (x_i - x_j)}{2\sigma^2}\right)$, where σ^2 denotes the variance to be defined by the user. Using the Kernel, $f(x; w)$ can be rewritten as $f(x; w) = \sum_{i=1}^{N} \beta_i y_i K(x, x_i) + b$.

For the training samples along the decision boundary, the corresponding α_is are greater than zero, as ascertained by the Karesh-Kuhn-Tucker theorem. These samples are known as support vectors. The number of support vectors is generally much smaller than N, being proportional to the generalization error of the classifier. A test vector $x \in R^m$ is then assigned to a given class according to $f(n) = sign[\mathbf{w}^T \phi(\mathbf{x}) + b] = sign(\sum_{i=1}^{N} \alpha_i y_i K(\mathbf{x}, \mathbf{x_i}) + b)$

E. Extreme Learning Machine

Huang et al. [7], [11] proposed a novel machine learning algorithm called the Extreme Learning Machine (ELM) that has significantly faster learning speed and requires less human intervention than other learning methods. It has been proven that the hidden nodes of the "generalized" single-hidden-layer feedforward networks (SLFNs) can be randomly generated and that the universal approximation capability of such SLFNs can be guaranteed. The ELM can determine analytically all the parameters of SLFNs instead of adjusting parameters iteratively. Thus, it can overcome the demerits of the gradient-based method and of most other learning methods. Compared to the most effective SVM-based method, the latest research [11] also shows that the ELM tends to achieve better generalization performance, less sensitivity to user-specified parameters, and easier implementation than a traditional SVM.



(a) AES2501 (b) FPR620 (c) FT-2BU

(d) URU4000 (e) ZY202-B

Figura 1: Examples of scanners images

Tabela I: Information about sensors

Sensor Model	Size
AES2501	Not fixed
FPR620	256×304
FT-2BU	152×200
URU4000	294×356
ZY202-B	400×400

IV. Computational Experiments

In what follows, we provide details about the dataset used in the experiments and how the experiments were set up. Then, we present the accuracy results revealed by the classifiers, considering the FingerCode as feature extractor. In this paper, the one-versus-one approach was adopted when using the SVM and Neural Networks.

A. SDUMLA-HMT Database

For assessing the performance of the classifiers in the task of biometric recognition, we have employed the SDUMLA-HMT Database made available to research community through [1]. SDUMLA-HMT was collected during the summer of 2010 at Shandong University, Jinan, China. This database consists of the following biometric modality: face, finger vein, gait, iris and fingerprint of 106 individuals (including 61 males and 45 females with age between 17 and 31,). In this paper, we use only the images of the fingerprint.

The fingerprint images on SDUMLA-HMT database [25] are collected with five different sensors (multi-sensor database, see table I for more details), the figure 1 shows examples of fingerprints collected by the scanners. Fingerprint images

[1] http://mla.sdu.edu.cn/sdumla-hmt.html

Tabela II: Results obtained for the fingerprint recognition using Neural Networks.

Number of Neurons	AES2501	FPR620	FT-2BU	URU4000	ZY202-B
	$\mu(\%) \pm \sigma(\%)$	$\mu(\%) \pm \sigma(\%)$	$\mu(\%) \pm \sigma(\%)$	$\mu(\%) \pm \sigma(\%)$	$\mu(\%) \pm \sigma(\%)$
2	41.49 ±2.83	74.84 ±2.53	62.89±3.57	67.19±3.29	40.62±2.24
5	43.92±2.65	76.48 ±2.63	64.17±3.33	68.75±2.64	42.86±1.85
10	44.45±2.80	77.46 ±2.22	64.90±3.17	69.54±2.82	43.88±2.11
15	45.15±2.84	78.15 ±2.22	65.44±3.41	70.06±2.63	44.44±2.17
20	44.99±2.61	78.38 ±2.17	65.54±3.17	70.02±2.46	44.53±1.97
30	45.2 ±2.63	78.50 ±2.36	65.67±3.68	70.14±2.43	44.38±2.06

Tabela III: Results obtained for the fingerprint recognition using Support Vector Machines.

σ	**AES2501**	**FPR620**	**FT-2BU**	**URU4000**	**ZY202-B**
	Mean(%)± Std (%)	Mean(%)± Std(%)	Mean(%)± Std(%)	Mean(%)± Std(%)	Mean(%)± Std(%)
2^3	49.03 ±3.43	74.18 ±3.10	59.01 ±4.49	69.90 ±3.31	48.20 ±1.75
2^4	52.01 ±2.90	79.26 ±2.09	67.93 ±3.93	72.51 ±2.83	51.04 ±1.45
2^5	49.80 ±2.88	79.47 ±2.39	67.77 ±3.65	72.11 ±2.77	48.62 ±1.46
2^6	46.17 ±2.75	78.35 ±2.27	65.65 ±3.86	70.57 ±2.78	45.25 ±1.82
2^7	44.47 ±2.53	77.73 ±2.17	64.62 ±3.94	69.67 ±2.82	43.81 ±1.62
2^8	43.77 ±2.54	77.11 ±2.42	63.85 ±3.71	68.89 ±2.88	43.01 ±1.74
2^9	37.85 ±2.38	73.65 ±2.09	57.11 ±3.54	63.58 ±2.60	36.81 ±1.46
2^{10}	23.59 ±1.56	54.61 ±2.27	38.58 ±1.72	43.22 ±1.51	23.48 ±1.15

Tabela IV: Results obtained for the fingerprint recognition using Optimum Path Forest.

Distances	**AES2501**	**FPR620**	**FT-2BU**	**URU4000**	**ZY202-B**
	Mean(%)± Std (%)	Mean(%)± Std(%)	Mean(%)± Std(%)	Mean(%)± Std(%)	Mean(%)± Std(%)
Euclidean	50.51 ±3.09	76.06 ±2.72	58.90 ±3.63	71.23 ±2.60	49.64 ±1.39
Chi-Square	46.11 ±2.70	72.81 ±2.15	54.20 ±3.77	69.54 ±3.02	45.67 ±1.93
Manhattan	49.94 ±2.66	77.87 ±1.84	60.17 ±3.25	72.49 ±2.54	49.39 ±1.34
Canberra	42.92 ±2.14	69.42 ±2.14	47.06 ±3.19	68.44 ±2.67	42.18 ±1.49
SquaredChord	46.60 ±2.72	72.35 ±2.24	53.92 ±3.47	69.26 ±2.59	45.82 ±1.76
SquaredChi-Squared	47.48 ±2.91	73.11 ±2.47	55.16 ±3.40	69.91 ±2.71	46.69 ±1.61
BrayCurtis	42.92 ±2.14	69.42 ±2.14	47.06 ±3.19	68.44 ±2.67	42.18 ±1.49

Tabela V: Results obtained for the fingerprint recognition using K-nearest neighbors.

K	**AES2501**	**FPR620**	**FT-2BU**	**URU4000**	**ZY202-B**
	Mean(%)± Std (%)	Mean(%)± Std(%)	Mean(%)± Std(%)	Mean(%)± Std(%)	Mean(%)± Std(%)
1	50.53 ±2.98	76.77 ±2.53	58.97 ±3.40	71.64 ±2.75	49.76 ±1.45
3	43.13 ±2.52	71.29 ±2.27	49.94 ±3.01	66.54 ±2.54	42.82 ±1.19
5	41.21 ±2.24	70.42 ±2.18	48.67 ±3.09	65.23 ±2.86	41.15 ±0.90
7	39.74 ±2.00	68.88 ±1.78	47.18 ±2.96	62.74 ±2.96	39.57 ±0.96

in SDUMLA-HMT database are acquired from six fingers such as: thumb finger, index finger and middle finger, of both hands. Notice that SDUMLA-HMT Group has requested from participants eight impressions (attempts) for each of six fingers to five previous mentioned sensors.

We consider all images from each sensor as a database and we have checked the quality of images from five different sensors (for each of these databases). It can be noticed that the best quality and the worst quality databases are those generated by FPR620 and FT-2BU sensors respectively

B. Experimental Setup

The SVM is a binary classifier that can not be applied directly to a multi-class problem, thus to overcome this problem we used the one-vs-one strategy. Moreover, the efficiency and effectiveness of the SVM training process depend directly on the a priori selection of the values of some control parameters. One of them, denoted by C, controls the tradeoff between margin maximization and error minimization. Other parameters appear in the non-linear mapping into feature space (the Kernel and its parameters, denoted by σ). Although we know that there are several rules of thumb to select the value of the kernel parameters [5], in this paper, we opted for using values of σ as 2^i, with $i = \{-2, -1, ..., 14, 15\}$, and chose empirically keeping the parameter C constant in 1000. This value for C was achieved after some preliminary experiments and agrees with the fact that SVM models with low values of C tend in general to attain better performance than those with high values of this parameter.

Regarding the OPF classifier, we use the following distance measures: Euclidean, Chi-Square, Manhattan, Canberra, SquaredChord, SquaredChi-Squared, BrayCurtis. There is no rule to select the optimum number of neurons in the hidden layer of a Neural Network. However, some thumb rules are available for calculating number of neurons. In this work, we have opted to set the values of number of neurons as 2, 5, 10, 15, 20, 30. Whereas, for the k-NN classifier the value of K was set as 1, 3, 7 and the distance measure used was Euclidean distance. For the ELM classifier, the number of neurons was

Tabela VI: Results obtained for the fingerprint recognition using Extreme Learning Machine.

Number of Neurons	**AES2501**	**FPR620**	**FT-2BU**	**URU4000**	**ZY202-B**
	Mean(%)± Std (%)	Mean(%)± Std(%)	Mean(%)± Std(%)	Mean(%)± Std(%)	Mean(%)± Std(%)
30	34.73 ±2.63	71.66 ±2.46	53.45 ±3.85	63.03 ±2.91	34.58 ±1.63
50	38.11 ±2.63	74.28 ±2.54	58.49 ±4.00	65.67 ±3.36	37.41 ±1.79
100	39.50 ±2.25	75.67 ±2.27	60.61 ±3.23	67.15 ±3.07	39.27 ±1.85
200	42.43 ±2.55	77.21 ±2.38	63.46 ±3.27	69.13 ±2.67	42.19 ±2.18
300	43.25 ±2.80	77.96 ±2.21	64.42 ±3.64	69.35 ±2.60	42.81 ±1.68

set as 30, 50, 100, 200, 300. The variation of the parameters values was performed in order to find the best parameter for the problem addressed here. For each of values chosen, a 10-fold cross-validation process was performed in order to better measure the average performance of the classifiers. Moreover, the datasets were scaled using min-max normalization.

C. Simulation Results

Tables II, III, IV, V and VI provides the best results obtained in the experiments, in terms of identification error achieved by classifiers different, when they were induced with the coefficients generated by FingerCode. The accuracy is shown in terms of average and standard deviation of the cross-validation error rate for the best-calibrated classifier. The error rate was calculated as the number of missclassifications divided by the total number of test examples.

Analyzing Tables II, III, IV, V and VI , it can be noted that the best result was achieved with the FPR620 sensor. The worst result was obtained with the AES2501 sensor. This demonstrates that the quality of fingerprint images is very important for biometric recognition.

The results obtained with Neural Networks for the AES2501 and ZY202-B sensors were very similar, except for the number of neurons equal to 2. The best result was obtained for the sensor FPR620 with 30 neurons. In this case, a recognition rate of 78.50 ± 2.46 was obtained. Regarding the variation of the number of neurons, it can be observed that this produced small variation in the performance.

For all sensors, the SVM produced the best performance when compared to other classifiers. In this case, the optimal sigma value lies in the range $[2^4 - 2^5]$. It is important to note that performance degrades rapidly to other σ values. This indicates that the paramenter's selection for the kernel is very important.

With the exception of the AES2501 sensor, the best result with the OPF was obtained using Manhattan distance. For the FPR620 sensor, a recognition rate of 77.87 ± 1.84 was obtained. It is possible to note that different distance metrics did not produce a great variation in the performance of the OPF. From the results, it is possible to conclude that the chosen of the distance metric is not a very important factor to be taken into account.

With regard to kNN, the best result was achieved with $k = 1$ for all sensors. Values of k greater than 5 produce a degradation in kNN performance when using the AES2501 and ZY202-B sensors.

Regardless of the sensor, ELM produced the best result with 300 neurons. The best recognition rate was obtained using FPR620 sensor. In this case, a recognition rate of 77.96 ± 2.21 was obtained.

D. Hypothesis test

Tabela VII: Wilcoxon test for the scanner between SVM

	Scanner	Classifier	Parameter	P value	Null hypothesis
SVM with σ = 16	AES2501	MLP	30	0.0002	Rejected
		OPF	Euclidian	0.0001	Rejected
		KNN	1	0.0002	Rejected
		ELM	300	0.0006	Rejected
SVM with σ = 16	FPR620	MLP	30	0.3066	Not rejected
		OPF	Manhatann	0.1194	Not rejected
		KNN	1	0.0252	Rejected
		ELM	300	0.1116	Not rejected
SVM with σ = 16	ZY202-B	MLP	30	0.0002	Rejected
		OPF	Euclidian	0.0814	Not rejected
		KNN	1	0.0957	Not rejected
		ELM	300	0.0002	Rejected
SVM with σ = 16	FT-2BU	MLP	30	0.1038	Not rejected
		OPF	Manhatann	0.0028	Rejected
		KNN	1	0.0006	Rejected
		ELM	300	0.0450	Rejected
SVM with σ = 16	URU4000	MLP	30	0.0376	Rejected
		OPF	Manhatann	0.7051	Not rejected
		KNN	1	0.2119	Not rejected
		ELM	300	0.0310	Rejected

In this work we used the test de Wilcoxon [14] as a mode hypothesis test. The Wilcoxon test is a non-parametric method for the results of two paired samples. At first they are calculated with the numerical values of the difference between each pair. In order to compile the Wilcoxon test we used the best classifiers and their parametrization consequences, a null hypothesis to be tested was that the classifiers were the same.

After the analysis of the hypothesis test and its accuracy of classification, the hypothesis test was analyzed in order to verify if the classifier SVM in order to verify if this classifier was really the best classifier for the five scanners presented. The table VII present all results after analysis of the SVM classifier for the other classifiers presented in this work. The AES2501 scanner presents the results of the classifiers OPF, MLP, ELM and KNN in relation to SVM, and it was seen that the SVM overcame all other classifiers. With the FPR620 scanner is clear that the SVM has only exceeded the KNN. we can see that the SVM did not present better results than the MLP neural network according to the digital images collected with the help of the FT-2BU scanner. The MLP and ELM were superseded by the SVM with the URU4000 scanner. In the ZY202-B scanner, it is possible to realize that the SVM was able to obtain better results than the MLP and ELM classifiers, this conclusion being shown in table VII.

V. Concluding Remarks

In this work, we have provided a assessment of the performance of several classifiers when coping with the task biometric recognition based on fingerprint. For this purpose, FingerCode was adopted for data preprocessing.

Overall, the results show that all classifiers achieved high recognition rates. It can be observed that the performance achieved by the different classifiers is very similar. The factor that most influenced the performance of the classifiers was the quality of fingerprint images. On the other hand, with the exception of SVM and k-NN, the variation of the classifiers parameters did not produce a great variation in the classifiers performance. From the results obtained by analyzing the accuracy rate and the Wilcoxon test it is possible to notice that the SVM was the classifier that produced better performance when compared to the other classifiers. It can be seen that the SVM for all sensors was better than at least one of the classifiers when applied to the five sensors. This classifier has obtained superior results to the other techniques presented in this work with the aid of the AES2501 Scanner, As shown in the table VII.

As ongoing work, we are currently extending the scope of investigation by considering other SVM configured with other kernel functions (and parameters), other types of vector machines, such as the Proximal SVMs, the Lagrangian SVMs [15], as well as (and most importantly) the conjoint influence of the hyper-parameters. In the future, we plan to investigate how the combination of models coming from different types of vector machines, each configured with the same values of the control parameters, can improvethe levels of performance, in terms of accuracy and generalization, from that achieved by each vector machine type alone.

Referências

[1] A. Askarunisa, S. K, S. R. Liu, and S. M. Batcha. Finger print authentication using neural networks. *MASAUM Journal od Computing*, 1(2), 2009.

[2] R. Cappelli, A. Lumini, D. Maio, and D. Maltoni. Fingerprint classification by directional image partitioning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(5):402–421, May 1999.

[3] J.-H. Chang and K.-C. Fan. A new model for fingerprint classification by ridge distribution sequences. *Pattern Recognition*, 35(6):1209 – 1223, 2002.

[4] D. Cheng, G. Liu, and Y. Qiu. Applications of particle swarm optimization and k-nearest neighbors to emotion recognition from physiological signals. In *2008 International Conference on Computational Intelligence and Security*, volume 2, pages 52–56, Dec 2008.

[5] V. Cherkassky and Y. Ma. Practical selection of {SVM} parameters and noise estimation for {SVM} regression. *Neural Networks*, 17(1):113 – 126, 2004.

[6] M. M. Chong, H. N. Tan, L. Jun, and R. K. Gay. Geometric framework for fingerprint image classification. *Pattern Recognition*, 30(9):1475 – 1488, 1997.

[7] G. Feng, G. B. Huang, Q. Lin, and R. Gay. Error minimized extreme learning machine with growth of hidden nodes and incremental learning. *IEEE Transactions on Neural Networks*, 20(8):1352–1357, Aug 2009.

[8] G. M. Freitas, A. M. H. Avila, J. P. Papa, and A. X. Falcao. Optimum-path forest-based rainfall estimation. In *2009 16th International Conference on Systems, Signals and Image Processing*, pages 1–4, June 2009.

[9] E.-H. S. Han, G. Karypis, and V. Kumar. *Text Categorization Using Weight Adjusted k-Nearest Neighbor Classification*, pages 53–65. Springer Berlin Heidelberg, Berlin, Heidelberg, 2001.

[10] S. Haykin. *Neural Networks and Learning Machines*. Number v. 10 in Neural networks and learning machines. Prentice Hall, 2009.

[11] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew. Extreme learning machine: Theory and applications. *Neurocomputing*, 70(1–3):489 – 501, 2006. Neural NetworksSelected Papers from the 7th Brazilian Symposium on Neural Networks (SBRN '04)7th Brazilian Symposium on Neural Networks.

[12] A. K. Jain, S. Prabhakar, and L. Hong. A multichannel approach to fingerprint classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(4):348–359, Apr 1999.

[13] J. Kacur, R. Vargic, and P. Mulinka. Speaker identification by k-nearest neighbors: Application of pca and lda prior to knn. In *2011 18th International Conference on Systems, Signals and Image Processing*, pages 1–4, June 2011.

[14] J. a. LITCHFIELD and F. Wilcoxon. A simplified method of evaluating dose-effect experiments. *Journal of pharmacology and experimental therapeutics*, 96(2):99–113, 1949.

[15] O. L. Mangasarian and D. R. Musicant. Lagrangian support vector machine classification. Technical Report 00-06, Data Mining Institute, Computer Sciences Department, University of Wisconsin, Madison, Wisconsin, June 2000. ftp://ftp.cs.wisc.edu/pub/dmi/tech-reports/00-06.ps.

[16] K. A. Nagaty. Fingerprints classification using artificial neural networks: a combined structural and statistical approach. *Neural Networks*, 14(9):1293 – 1305, 2001.

[17] J. P. Papa, A. X. Falcao, A. L. M. Levada, D. C. Correa, D. H. P. Salvadeo, and N. D. A. Mascarenhas. Fast and accurate holistic face recognition using optimum-path forest. In *2009 16th International Conference on Digital Signal Processing*, pages 1–6, July 2009.

[18] J. P. Papa, A. X. Falcão, and C. T. N. Suzuki. Supervised pattern classification based on optimum-path forest. *International Journal of Imaging Systems and Technology*, 19(2):120–131, 2009.

[19] J. P. Papa and A. Rocha. Image categorization through optimum path forest and visual words. In *2011 18th IEEE International Conference on Image Processing*, pages 3525–3528, Sept 2011.

[20] J. P. Papa, A. A. Spadotto, A. X. Falcao, and J. C. Pereira. Optimum path forest classifier applied to laryngeal pathology detection. In *2008 15th International Conference on Systems, Signals and Image Processing*, pages 249–252, June 2008.

[21] C. H. Park and H. Park. Fingerprint classification using fast fourier transform and nonlinear discriminant analysis. *Pattern Recognition*, 38(4):495 – 503, 2005.

[22] M. Sarkar and T. T. Leong. Application of k-nearest neighbors algorithm on breast cancer diagnosis problem. In *Proceedings of the AMIA Annual Symposium, Los Angeles, USA*, 2000.

[23] B. Scholkopf and A. J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, Cambridge, MA, USA, 2001.

[24] V. N. Vapnik. *The nature of statistical learning theory*. Springer-Verlag New York, Inc., New York, NY, USA, 1995.

[25] Y. Yin, L. Liu, and X. Sun. Sdumla-hmt: A multimodal biometric database. In Z. Sun, J. Lai, X. Chen, and T. Tan, editors, *Biometric Recognition*, volume 7098 of *Lecture Notes in Computer Science*, pages 260–268. Springer Berlin Heidelberg, 2011.

[26] X. P. Yu and X. G. Yu. Novel text classification based on k-nearest neighbor. In *2007 International Conference on Machine Learning and Cybernetics*, volume 6, pages 3425–3430, Aug 2007.