Using PCA to Recognize Characters in Vehicle License Plates

Luiz Reginaldo A. F. Curado *
Electrical Engineering Dept.
University of Brasilia
Brasília, Brasil

Adolfo Bauchspiess
Electrical Engineering Dept.
University of Brasilia
Brasília, Brasil

Valquíria B. F. Curado
Electrical Engineering Dept.
University of Brasilia
Brasília, Brasil

Abstract - *This paper considers the approach of the principal component analysis for character classification in a vehicle license plate recognition system. The presentation of the method is generic in the sense that it also contemplates the possibility of use in other kinds of problems involving pattern recognition.*

Keywords: Vehicle license plate recognition, character recognition, principal component analysis.

1 Introduction

The ability to recognize characters is one of the problems envolved in the design of a vehicle license plate recognition system, and several approaches has been proposed to this specific task, as neural networks [1,2,3] and template matching [4,5,6,7].

In a previous paper[8], we presented a system that uses principal component analysis (PCA) to perform the character recognition task. PCA is a well-know method in works that deals with face recognition [9, 10], and consists into obtain a transformation in wich the initial set of samples can be represented using a reduced number of significant characteristics, holding still the majority of the content of information of the data [11].

In this work, the character recognition process performed by our system is analyzed in more detail. We present the PCA approach to the recognition task, followed by the experimental results obtained with this method. For evaluation, we selected a database containing 780 vehicles images, in wich the position of the characters were correctly obtained by our plate recognition system.

As contribution of this work, we state that the PCA approach could also be used to solve general pattern recognition problems. The results obtained in experimental part comproves this idea.

2 Plate and character recognition

As stated in a previous work [8], the problem of vehicle license plate recognition can be decomposed in three more specific problems: find the license plate, locate and recognize the plate characters. These tasks are illustrated in Figure 1.

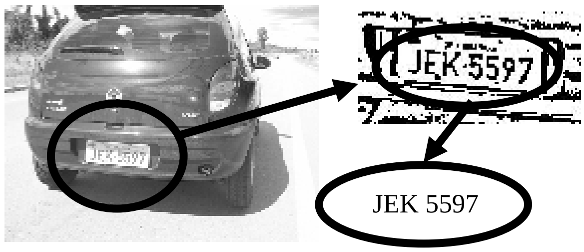

Figure 1. Tasks related to license plate recognition.

In this paper we focus the problem of character recognition. A previous work [8] presented the approaches used in solving the other problems.

Much of the previous researches on character recognition in vehicle license plate systems have used neural networks [1,2,3] and pattern matching [4,5,6,7] to perform the recognition task. In our system, we followed a different approach, and used the PCA approach to classify the plate characters.

Our choice was motivated by the results obtained with works that deals with face recognition [9, 10].

3 The PCA method

The PCA method can be used to extract relevant information from data sets, and consists of looking for another basis that better express the distribution of the data. This basis is obtained by means of a linear combination of the original basis, and removes part of the redundancy of the data [12].

PCA is useful to descorrelate data, performing a transformation that projects the data in the orthogonal directions of bigger variance, named the principal components of the data. PCA is also a technique used to obtain a more compact encoding of the data, allowing reduce its dimension from n to p, where $p \leq n$.

The compression operation is explained by the property that usually great part of the variance is given by a reduced number of components. Thus, using PCA, it can be discarded the components related to low variance in the data, without great loss of information. In fact, it can be proved that PCA provides an optimal technique for linear dimension reduction, considering average quadratic error [11].

3.1 Change of basis

Let X and Y be n x m size matrices, where X denotes an initial set of experimental samples, and Y is the set obtained by performing on X a linear transformation P such that:

$$Y = PX \tag{1}$$

Then Y denotes a new way to express the sample set X, by means of a basis change, where the lines of P, defined by the set $\{p_1, ..., p_n\}$ are a new set of base vectors to express the columns of X. This property can be observed analyzing equation 1 in more detail:

$$PX = \begin{bmatrix} p_1 \\ \cdots \\ p_n \end{bmatrix} \begin{bmatrix} x_1 & \cdots & x_m \end{bmatrix}$$

$$= \begin{bmatrix} p_1x_1 & \cdots & p_1x_m \\ \cdots & \cdots & \cdots \\ p_nx_1 & \cdots & p_nx_m \end{bmatrix} \tag{2}$$

In this transformation, each column Y_i of Y is obtained from the product of the column X_i of X with the vectors $\{p_1, ..., p_n\}$, for $1 \le i \le m$. It can be said, then, that each column Y_i is a projection of the vector X_i in the basis $\{p_1, ..., p_n\}$. In PCA approach, $\{p_1, ..., p_n\}$ are the principal components of X.

3.2 Covariance and redundancy

One way to quantify the existing correlation between individual samples consists in calculating the covariance of the data. The covariance σ^2_{ab} between two samples $a = [a_1, a_2, ..., a_n]$ and $b = [b_1, b_2, ..., b_n]$, in turn, is defined as:

$$\sigma^2_{ab} = \frac{\sum_{i=0}^{n}(a_i - \bar{a})(b_i - \bar{b})}{n-1} \tag{3}$$

Where $\bar{a}$ and $\bar{b}$ express the mean of the samples a and b, respectively. Having the samples zero mean, equation 3 then becomes:

$$\sigma^2_{ab} = \frac{\sum_{i=0}^{n} a_i b_i}{n-1} = \frac{1}{n-1} ab^T \tag{4}$$

As this measure is calculated by taking two data samples, in analysis problems that involves more than two samples it is necessary to calculate the covariance between each pair of samples.

Let m be the number of samples taken for analysis. Then, the covariance matrix C, with size m x m, is defined as:

$$C_{ij} = \sigma^2_{ij} \tag{5}$$

Where C_{ij} is the element placed in row i and column j of the matrix, and σ^2_{ij} is the covariance between the samples i and j. The covariance matrix is a square shaped and symmetrical matrix.

Now, let's assume that matrix X keeps in its columns a set of m samples, with zero mean. The covariance matrix of X, named C_x is then defined as:

$$C_x = \frac{1}{n-1} XX^T \tag{6}$$

The diagonal elements of C_x are related to the variance of the samples stored in the columns of matrix X. The off-diagonal elements, in turn, are related to the covariance between pairs of samples.

When covariance is high, we say that there is high redundancy in the considered data sets [12]. Then, in order to reduce the redundancy of the set of samples X, we would like that transformation (1) to be performed in such a way that each variable co-vary as little as possible with other variables.

In the ideal case the transformation must have the property that the covariance between distinct samples is zero, and the covariance matrix of Y, named C_y, is a diagonal matrix. With this condition true, the samples in Y will be then descorrelated.

Having this ideas in mind, we can state the objective of PCA as the process to obtain a transformation P such that Y=PX and C_y is a diagonal matrix.

3.3 Solving PCA

To solve the PCA problem, it can be used the eigenvector decomposition property. Then, analyzing the computation of the covariance matrix C_y, we have:

$$C_y = \frac{1}{n-1} YY^T$$

$$= \frac{1}{n-1} (PX)(PX)^T$$

$$= \frac{1}{n-1} PXX^T P^T$$

$$= \frac{1}{n-1} P(XX^T)P^T \tag{7}$$

Now, observe that matrix XX^T is symmetric:

$$(XX^T)^T = X^{TT}X^T = XX^T \tag{8}$$

Also, it can be proved that a symmetrical matrix is diagonalized by an orthogonal matrix of its eigenvectors. [12]. In fact, taking $A = XX^T$, D the diagonal matrix that stores the eigenvalues of A in its diagonal elements and E the matrix that contains the eigenvectors of A in its columns, we can write, as consequence of the eigenvector definition:

$$AE = ED$$

$$A = EDE^{-1} \qquad (9)$$

The eigenvectors of a symmetric matrix, in turn, are orthogonal [12]. To show this property, assuming that λ_1 and λ_2 are distinct eigenvalues of matrix A, regarding to eigenvectors e_1 and e_2, it can be written:

$$\lambda_1 e_1.e_2$$

$$= (\lambda_1 e_1)^T e_2$$

$$= (Ae_1)^T e_2$$

$$= e_1^T A^T e_2$$

$$= e_1^T Ae_2$$

$$= e_1^T (\lambda_2 e_2)$$

$$= \lambda_2 e_1.e_2 \qquad (10)$$

Then, we have that $(\lambda_1-\lambda_2)\, e_1.e_2 = 0$, but as $\lambda_1 \neq \lambda_2$, it follows that $e_1.e_2 = 0$ and E is an orthogonal matrix. Moreover, we have that $E^T = E^{-1}$, because:

$$(E^T E)_{ij} = \begin{cases} 1 \text{ when } i = j \\ 0 \text{ when } i \neq j \end{cases} \qquad (11)$$

As $E^T E = I$, the identity matrix, it follows that $E^{-1} = E^T$ Using this result in (9), we obtain:

$$A = EDE^T \qquad (12)$$

Finally, taking $P = E^T$, and using (12) in (7), we get:

$$C_y = \frac{1}{n-1} E^T (EDE^T) E$$

$$= \frac{1}{n-1} (E^T E) D (E^T E)$$

$$= \frac{1}{n-1} D \qquad (13)$$

Then, it can be concluded that the principal components of X are the eigenvectors of the covariance matrix XX^T. Using this result, the transformation performed by PCA is then:

$$Y = E^T X \qquad (14)$$

As consequence of this transformation, we have that the variance of the set Y is equal to the eigenvalues of C_x. The eigenvectors can then be thought of as a set of features that together characterize the variation between the samples stored in X.

The initial sample set X, in turn, can be obtained from Y, using the following transformation:

$$Y = E^T X$$

$$EY = EE^T X$$

$$EY = X \qquad (15)$$

If we are using all the eigenvectors of C_x to perform the transformation 14, then we can again obtain X without data loss. On the other hand, if not all eigenvectors are considered, X can will be rebuilt with data loss.

So, taking the eigenvectors related to the biggest eigenvalues, the vector X will be projected in the directions of bigger variance.

4 The approach for pattern recognition

PCA can be used to classify patterns, as pointed by some previous works in face recognition [9, 10]. We used this approach to classify characters in a license plate recognition system[8], and state that PCA can also be used in many others pattern recognition problems.

The process used to train a PCA classifier involves the following steps:

1) Define an initial set of training patterns. Each pattern P_i of the set, where $1 \leq i \leq m$, belongs to some of the classification classes. Also, we assume that P_i is a n x 1 vector.

2) Calculate the average vector of the training set. This vector, named Ψ, will defined as:

$$\Psi = \frac{1}{m} \sum_{i=1}^{m} P_i \qquad (16)$$

The obtained Ψ vector has then size n x 1.

3) From the initial set of training patterns, define matrix A (the pattern space), putting in each of its columns one of the m training patterns, subtracted from it the average vector Ψ:

$$A = [\, P_1 - \Psi \quad P_2 - \Psi \quad \ldots \quad P_m - \Psi \,] \qquad (17)$$

The obtained A matrix has then size n x m.

4) Calculate the eigenvectors of the covariance matrix of the set of training patterns. This matrix, named C, will be defined as:

$$C = AA^T \tag{18}$$

The covariance matrix has then size n x n, where n is the size of one training pattern. In our specific problem, each pattern is an image of size 40 x 40 = 1.600 pixels, and the covariance matrix has, then, 1.600 x 1.600 data elements. As the size of C is big, usually the matrix L is considered [9]:

$$L = A^T A \tag{19}$$

The L matrix has size m x m, where m is the number of training patterns. It can be proved, in turn, that the eigenvectors of C can be obtained from the eigenvectors of L, by means of a linear combination of A and the eigenvectors of L [9]. In fact, assuming that v_i and λ_i are eigenvectors and eigenvalues of L, respectively, we have:

$$Lv_i = \lambda_i v_i$$

$$(A^T A)v_i = \lambda_i v_i$$

$$A(A^T A)v_i = A\lambda_i v_i$$

$$(AA^T)Av_i = \lambda_i Av_i$$

$$CAv_i = \lambda_i Av_i \tag{20}$$

Equation 20 shows that Av_i is an eigenvector of the matrix $C = AA^T$. So, let V be the matrix that contains in its columns the m eigenvectors of L. The U matrix, that contains in its columns the eigenvectors of C, can then be obtained calculating:

$$U = AV \tag{21}$$

As matrix V has size m x m, and matrix A has size n x m, it follows that matrix U has size n x m.

5) Select p eigenvectors of C to compose the principal component matrix, where $p \leq m$. The selected eigenvectors are columns of matrix U related to the biggest eigenvalues found in computating the eigenvectors of the covariance matrix. The selection is done in order to keep in matrix U only the selected p eigenvectors:

$$U = [\ U_1 \quad U_2 \quad \ldots \quad U_p\ ] \tag{22}$$

As each eigenvector has n elements and it was selected p eigenvectors, the final U matrix has then size n x p.

6) Project each training pattern P_i onto the selected eigenvectors space, in order to obtain a new encoding Ω_i of the pattern:

$$\Omega_i = U^T (\ P_i - \Psi\ ) \tag{23}$$

The obtained encoded Ω_i vector has then size p x 1.

From the steps suggested by the PCA training algorithm, step 5 is especially important, because it allows reduce the size of the training patterns from n x 1 to p x 1. This step can then be seen as a compression step.

Let P_{test} be a n x 1 size pattern submitted to the classification task. To perform this work, the classifier executes the following operations:

1) Subtract the average vector Ψ from P_{test}. Then, project the obtained vector onto the eigenvectors space U, in order to obtain a new encoding Ω_{test} of the pattern:

$$\Omega_{test} = U^T (P_{test} - \Psi) \tag{24}$$

The obtained Ω_{test} vector has then size p x 1.

2) Analyze the encoded training patterns Ω_i, in order to find the Ω pattern such that:

$$\Omega = \min \|\ \Omega_{test} - \Omega_i\ | \tag{25}$$

The submitted pattern P_{test} is then classified as belonging to the same class that X_i if Ω_i is the obtained pattern in applying the Euclidean distance, as stated by equation 25.

5 Experimental Results

To the problem of character recognition, two PCA classifiers were constructed: one for digit classification and another for letter classification. We used a sample set containing 780 brazilian vehicles images, from which 300 were selected to train the classifiers, and 480 were used to validation. Of the images selected for training, 2.084 characters were used, and of the images selected for classification 3.313 characters were used. Table 1 presents this data.

Table 1. Selection of the sample set data.

Training		Classification	
Images	Characters	Images	Characters
300	2.084	480	3.313

The position of each character in the selected images was obtained during the process of analysis and segmentation of the vehicle plate. Then, for each character a rectangular region of pixels regarding to the character image was extracted.

On this region an adaptative binarization algorithm was applied [13], and the region was rescaled to 40x40 pixels. This image was then placed in a 40 x 40 array, and submitted to the respective classifier. The performance of each classifier is presented in Table 2.

Table 2. Classifier performance.

Character	Samples	Sucess	Correct
Letter	1.422	1.359	95,56%
Digit	1.891	1.850	97,83%

Finally, we compare in Table 3 the performance of our system with those presented into literature. As can be observed, the obtained results were satisfactory.

Table 3. Performance of different systems.

	Character Classification
SIAV[1]	91,30%
Barroso [4]	98,40%
Kwasnicka [5]	95,24%
Shyang-Lih [7]	95,60%
Our system	96,70%

The system was implemented using the Java language, and the tests were taken in a Pentium II, 2.2 GHz.

6 Conclusions

The objective of this work was to show that PCA can be used to build classifiers in problems related to pattern recognition. In particular, the technique was used to perform character classification in a vehicle license plate recognition system.

The results obtained in the experimental part proves the objective of the work. The simplicity of the method, in turn, becomes its use attractive.

7 References

[1] Souza, F. P. C. and Susin, A. "SIAV – Um Sistema de Identificação Automática de Veículos". CBA2000.

[2] Maged M. M. Fahmy. Automatic number-plate recognition: neural network approach. In Proceedings of Vehicle Navigation and Information Systems Conference, 1994.

[3] Kwaśnicka H., Wawrzyniak B. Licence Plates Localisation and Recognition in camera pictures. In 3rd Symposium on Methods of Artificial Intelligence, 2002.

[4] J. Barroso, A. Rafael, E. L. Dagless, J. Bulas-Cruz. Number plate reading using computer vision. In International Symposium on Industrial Electronics, 1997.

[5] Muhammad Sarfraz, Mohammed Jameel Ahmed, Syed A. Ghazi. Saudi Arabian License Plate Recognition System. In International Conference on Geometric Modeling and Graphics, 2003.

[6] D. Irecki, D. G. Bailey. Vehicle registration plate localization and recognition. In Proceedings of the Electronics New Zealand Conference, 2001.

[7] Shyang-Lih Chang; Li-Shien Chen; Yun-Chung Chung; Sei-Wan Chen. Automatic license plate recognition. In Intelligent Transportation Systems, 2004.

[8] Curado, L. R. A. F.; Bauchspiess, A.; Curado, V. B. F. "An Incremental Learning Vehicle License Plate Recognition System". In: The 2005 International Conference on Computer Vision, Las Vegas, EUA.

[9] Turk, M. and Pentland, A., "Eigenfaces for Recognition", In: Journal of Cognitive Neuroscience, Volume 3, Number 1, pages 71 to 86, 1991.

[10] Silva, P. Q., Rosa, A. N. C. S., "Face Recognition Based on Symmetryzation". In: Proceedings of CSIT, 2002.

[11] S. Haykin, "Neural Network: A Comprehensive Foundation,", (2nd Ed.), Prentice Hall, 1999

[12] Shlens, J. "A Tutorial on Principal Component Analysis: Derivation, Discussion, and Singular Value Decomposition". Online Note: http://www.snl.salk.edu/~shlens/notes.html, 2006.

[13] Niblack W. An Introduction to Digital Image Processing. Prentice Hall, p.155-116, 1986.