ABSTRACT

Increased mobility and internationalization open new challenges to develop effective traffic monitoring and control systems. This is true for automatic license plate recognition architectures that, nowadays, must handle plates from different countries with different character sets and syntax. While much emphasis has been put on the license plate localization and segmentation, little attention has been devoted to the huge amount of samples that are needed to train the character recognition algorithms. Nevertheless, these samples are difficult to get when dealing with an international-wide scenario that involves many different countries and the related legislations. This paper reports a new algorithm for License Plate recognition, developed under a joint research funded by Autostrade per l’Italia S.p.A., the main Italian highways company. The research aimed at achieving improved recognition rates when dealing with vehicles coming from different European and nearby states. Extensive experimental tests have been performed on a database of about 7,000 images comprising License Plates picked up by portals spread nationally. The overall rate of correct classification is 98.1%.

Index Terms— Image processing, Character recognition

1. INTRODUCTION

Nowadays, many License Plate recognition systems exist. A noticeable weak point of such systems is their difficulty in recognizing plates of different nationalities. One of the principal reasons is the huge amount of learning samples needed to train the OCR algorithms. Unfortunately, when license plates from different countries must be recognized, it is difficult to get an acceptable number of examples (for each character of each different nation). It becomes therefore necessary to develop improved recognition techniques, to achieve satisfactory results by a limited number of character instances.

The most diffused License Plate recognition use supervised learning. Usually both not-parametric classifiers, for example Neural Networks, and parametric classifiers are used [1]. Parametric Classifiers in general assume that a predetermined family of pdf (for instance, Gaussian Multivariate) describes the probability that a given character belongs to a given class. The large number of parameters that must be estimated (for example Covariance Matrix, Mean Values), imposes the use of many character instances [2]. The problem is partially eased by using dimensionality reduction techniques (like Principal Component Analysis + Linear Discriminant Analysis and Independent Component Analysis). Nevertheless, the number of samples remains high when considering many different character sets. Similar problems arise if Neural Networks are used (for example Multilayer Feedforward Networks) [3], [4]. The number of neurons in the hidden layer is large and demands for a huge learning-set. This is why to develop alternative approaches is important for international License Plate recognition.

The idea we propose, springs out from the observation that the statistical variability of many license plate characters originates from well-known deterministic distortions, like: rotations, scale changes, motion blurs, translations. This knowledge allows implementing a set of Generative Models (GM) to produce many synthetic characters whose statistical variability is equivalent (for each class) to that showed by real samples. Thus a suitable statistical description of a large set of characters can be obtained by using only a limited set of images. We propose to embed such models into a classification algorithm to reach high performance even if trained with very few examples.

2. GENERATIVE MODELS FOR LICENSE PLATE CHARACTERS

In the proposed system we extract each character of each License Plate in the training set (the License Plate is
automatically located and segmented by a procedure similar to that proposed in [4]). Then each character image is normalized to a fixed dimension $h \times l$ (16x10 pixels in our implementation). The gray levels are rearranged in a feature vector $X = (x_0, x_1, \ldots, x_m)$, with $m = h \times l$, by concatenating the rows of the normalized image. Each feature vector is an instance of a stochastic multivariate variable of a certain class, characterized by an unknown distribution. Such distribution is described by estimating the Covariance Matrix. Suppose that $\{X_1, X_2, \ldots, X_n\}$ is the learning-set of the class under analysis, where each element $X_j$ is a vector $(x_{j1}, \ldots, x_{jm})^T$, while $m = h \times l$. If we mark the sample mean by $\mu = \frac{1}{n} \sum_{i=1}^{n} X_i$, then the sample Covariance Matrix $C$ is:

$$
\frac{1}{n-1} \sum_{i=1}^{n} (X_i - \mu)(X_i - \mu)^T
$$

The Covariance Matrix gives important information about the statistical distribution of gray levels. In particular, for a Gaussian Multivariate distribution, the Covariance Matrix completely specifies it. A graphic representation of the $j$-th column of the Covariance Matrix can be obtained by arranging the column values into an image of $h \times l$. If we repeat the same procedure for the whole set of columns of the Covariance Matrix, we get $l \times h$ images. Now it is possible to build another image by arranging each single image that represents a column of the Covariance Matrix, into an array $h \times l$. As an example Figure 1 represents such an array related to the Covariance Matrix obtained from the learning set of character "2". Some peculiarities are noticeable in the Figure:

1) inside each single image, the brighter and darker areas (that represent high positive and negative correlation values) spread over the whole image. This means that each pixel in the feature vector is strongly correlated to other pixels even at large distances.

2) pixels in proximity of edges show higher variability. This is easily seen by the high intensities (both positive and negative) that are clustered near the edges of the character "2".

An interpretation of this Covariance Matrix is that the learning-set of a class, comprises a single reference pattern subjected to some deterministic transformations, like: translations, rotations, contractions, expansions, blurring. In fact, such transforms in general produce: 1) high correlation among distant pixels; 2) high variations in the pixels near edges.

Now consider the set $T = \{T_e, T_s, T_b, T_a, T_t, T_r, T_l\}$ which relates to the following seven transformations: 1) horizontal translation, 2) vertical translation, 3) rotation, 4) horizontal expansion, 5) vertical expansion, 6) horizontal blur, 7) vertical blur. Each transformation can be represented through a mapping $T_i: \mathbb{R}^2 \times \mathbb{R} \to \mathbb{R}$ that associates a bidimensional pattern $f(x, y)$ and a scalar value $v$ (that defines the transformation) to a transformed pattern $f'(x, y)$, where $f'(x, y) = T_i[f, v](x, y)$. If $v$ is small, each mapping can be linearly estimated by:

$$
T_i(f, v) \approx f + \frac{\partial T_i}{\partial v}(f, 0) \cdot v
$$

It is to note that $\frac{\partial T_i}{\partial v}(f, 0)$ is in turn a bidimensional pattern that can be obtained from the pattern $f$, therefore:

$$
T_i(f, v) \approx f + g_i(f) \cdot v\quad \text{with}\quad g_i: \mathbb{R}^2 \to \mathbb{R}^2
$$

For example, if we consider a horizontal translation by $v$ along the $x$-axis, we have $T_1 \{f, v\}(x, y) = f(x+v, y)$ that can be linearly guessed (up to the first order), as:

$$
f(x + v, y) \approx f(x, y) + \frac{\partial f(x, y)}{\partial x} \cdot v,
$$

therefore, by comparison with equation (3), it turns out:

$$
g_1(f)(x, y) = \frac{\partial f}{\partial x}(x, y).
$$

Similar reasoning holds for the seven transformations, finally giving:

$$
g_1(f)(x, y) = \frac{\partial f}{\partial x}(x, y)\quad\text{horiz. transl.}\quad (4)
$$

$$
g_2(f)(x, y) = \frac{\partial f}{\partial y}(x, y)\quad\text{vert. transl.}\quad (5)
$$

$$
g_3(f)(x, y) = \frac{\partial f}{\partial y}(x, y)x - \frac{\partial f}{\partial x}(x, y)y\quad,\text{rotation}\quad (6)
$$

$$
g_4(f)(x, y) = \frac{\partial f}{\partial x}(x, y)x\quad,\text{horiz. expansion}\quad (7)
$$

$$
g_5(f)(x, y) = \frac{\partial f}{\partial y}(x, y)y\quad,\text{vert. expansion}\quad (8)
$$

$$
g_6(f)(x, y) = \frac{\partial^2 f}{\partial x^2}(x, y)\quad,\text{horiz. blur}\quad (9)
$$
As previously suggested, the learning-set $X_i = \{X_{i1}, \ldots, X_{in}\}$ relative to the $i$-th character, can be interpreted as a collection of instances of a single basic pattern $f(x, y)$, undergoing some deterministic transformations. Formally, the element $X_{ij}$ of the learning-set is given by $X_{ij} = T_j(f_i(x, y))$, where $T_j$ is the composition of a subset of $T$. By using the previous linear estimations (4) to (10) it results that, independently from the number of deterministic transformations involved and from the order of their application, the following estimation is always valid:

$$X_j \approx f_i + v_{j1} \cdot g_{1i} + v_{j2} \cdot g_{2i} + \ldots + v_{j7} \cdot g_{7i}$$

(11)

According to the previous formula, the whole learning set of characters that belong to a certain class, equals a set of instances (identically spread) of the following stochastic variable:

$$X = f + v_1 \cdot g_{1i} + v_2 \cdot g_{2i} + \ldots + v_7 \cdot g_{7i} + g$$

$$g^T \cdot g_{ij} = 0, \quad i = 1, \ldots, 7$$

(12)

where, $v_1, v_2, \ldots, v_7$ are stochastic scalars that can be reasonably considered independent with zero mean. As an example, refer to the variable $v_1$ that represents the horizontal displacement of the reference pattern of a certain class of characters. It is reasonable that the likelihood of a rightward displacement is equal to the chance of a leftward displacement of the same amount. Thus the average value of $v_1$ is zero. The term $g$ of equation (12) is a stochastic vector that represents the portion of the difference, between the single instance of the learning-set and the matching reference prototype, that cannot be explained by the 7 deterministic transformations previously assumed. It is reasonable to suppose that the vector $g$ has zero mean. Assuming these hypotheses, the mathematical expectation $E[X]$ of the variable $X$ equals $f_i$ (that is the original reference pattern for class $i$). In turn the expectation $E[X]$ can be estimated by the sample mean

$$\mu = \frac{1}{n} \sum_{i=1}^{n} X_i$$

therefore $\mu$ gives an estimate of the reference pattern $f_i$. For our purpose it is important to note that, given $f_i$, the values of $g_{1i}, g_{2i}, \ldots, g_{7i}$ can be easily estimated by using equations (4) to (10).

3. CLASSIFICATION ALGORITHM

The proposed classification algorithm comprises two phases: 1) parameters setting by the training samples; 2) character classification by embedded Generative Models. The details are described below.

3.1. Training

According to the previous paragraph, it turns out that equation (12) defines a procedure to create instances of the characters to be recognized, if the correct values of $g_{1i}, \ldots, g_{7i}$ are given for each character-class. The proper values are obtained during a preliminary supervised learning phase. The training procedure is carried out as follows: consider a certain character-class (say the $i$-th). The corresponding training samples are averaged to get the reference pattern $f_i$. Once the pattern $f_i$ has been earned it is substituted to $f$ in equations (4) to (10) while an edge enhancement operator (for example Sobel) is used to estimate the gradients. In this way it is possible to estimate the vectors $g_{1i}, \ldots, g_{7i}$. The training procedure repeats until all the character-classes have been processed and the related parameters have been obtained. It is important to note that the sample mean gives a stable estimate of each reference pattern $f_i$; even by using a relatively small number of samples. In the limiting case, even a single sample can be used to train the classifier, if such a sample is good enough (see below) to give a “single-shot” estimate of $f_i$.

3.2. Classification

Be $c$ the character to be classified. The distance (difference) between $c$ and the pattern $f_i$ that represents the $i$-th class, is given by the value of $g$ and can be estimated
by using the previous equation (12). To classify the character \( c \), we calculate the value of the residual \( g \) for each representative pattern of each character-class. Finally, character \( c \) will belong to the class that gives the minimum value of the residual.

In a more formal way the whole process can be described as follows: first the instances \( v_1, v_2, ..., v_7 \) of the stochastic variables in equation (11) are estimated. We mark by \( v'_{i1}, v'_{i2}, ..., v'_{i7} \) the estimates given by:

\[
\begin{bmatrix}
v'_{i1} \\
v'_{i2} \\
\vdots \\
v'_{i7}
\end{bmatrix} = M_i \cdot \text{pinv}(M_i) \cdot (c - m_i)
\]

In the previous equation, \( m_i \) stands for the reference pattern of the i-th character-class obtained by averaging the corresponding training samples. We mark by \( M_i \) the matrix \( \begin{bmatrix}
v'_{i1} \\
v'_{i2} \\
\vdots \\
v'_{i7}
\end{bmatrix} \). Pseudo inversion is used to account for the over determined set of equations. The procedure repeats until all characters have been considered. Then the generative models are embedded into the classifier to get the estimate of the residual for each class. In particular, given a new character \( c \) to be classified, the residual \( g'_{ci} \) for the i-th class is calculated as follows:

\[
g'_{ci} = \| (c - m_i) - M_i \cdot \text{pinv}(M_i) (c - m_i) \| = \| (c - m_i) - M_i \cdot \text{pinv}(M_i) (c - m_i) \|\]

where \( \text{pinv}(c) \) gives the pseudo-inverse matrix.

The previous calculation is repeated for all the other character-classes. Finally \( c \) is classified as belonging to the class \( \Omega_i \) resulting in the minimum value of the residual:

\[
c \in \Omega_i \mid g'_i = \min_k \{ g'_k \}
\]