RECOGNITION OF TEXT WITH KNOWN GEOMETRIC AND GRAMMATICAL STRUCTURE

Jan Rathouský
Department of Control Engineering, Faculty of Elec. Eng., Czech Technical University in Prague, Czech

Martin Urban
Eyedea Recognition, Prague, Czech Republic
Center for Applied Cybernetics, Faculty of Elec. Eng., Czech Technical University in Prague, Czech

Vojtěch Franc
Fraunhofer Institut FIRST IDA, Berlin, Germany

Keywords: Text Recognition, Sructured Support Vector Machines, License Plate Recognition.

Abstract: The optical character recognition (OCR) module is a fundamental part of each automated text processing system. The OCR module translates an input image with a text line into a string of symbols. In many applications (e.g. license plate recognition) the text has some a priori known geometric and grammatical structure. This article proposes an OCR method exploiting this knowledge which restricts the set of possible strings to a limited set of feasible combinations. The recognition task is formulated as maximization of a similarity function which uses character templates as reference. These templates are estimated by a support vector machine method from a set of examples. In contrast to the common approach, the proposed method performs character segmentation and recognition simultaneously. The method was successfully evaluated in a car license plate recognition system.

1 INTRODUCTION

Recognition of text in images is an important part of the pattern recognition field. Systems for text recognition are generally referred to as OCR (Optical Character Recognition) systems.

This article presents a method for OCR that makes use of the fact that many examined texts have a given structure that can be described by a common model. In other words, the text yields to some grammar and layout, determining the number of symbols, their relative width and position and also the kind of symbols that can appear in each position. The advantage of this approach is that using the a priori knowledge about the text structure reduces the number of possible configurations, thus improving the success rate of the method, especially when the input image is in a bad quality. Typically the method fits for recognition of short structured texts (see Figure 1) taken in low resolution and possibly inappropriate light conditions.

The text recognition itself consists of two subtasks – the text segmentation, where areas (segments) of the image containing single characters are found and the text recognition, where the characters in individual segments are determined. The classical approach is to perform these subtasks separately, which leads to recognition errors if the segmentation is done incorrectly. Systems using this approach have

(a) license plate

(b) license plate (c) ADR/RID plate

Figure 1: Examples of images with short structured texts with a priori known geometrical and grammatical structure.

Rathouský J., Urban M. and Franc V. (2008).
RECOGNITION OF TEXT WITH KNOWN GEOMETRIC AND GRAMMATICAL STRUCTURE.
In *Proceedings of the Third International Conference on Computer Vision Theory and Applications*, pages 194-199
DOI: 10.5220/0001086501940199

been proposed for example in (Shapiro and Gluhchev, 2004; Ko and Kim, 2003; Lee et al., 2004) or (Rahman et al., 2003). A different approach, used also in the proposed method is to perform both operations at once, thus treating the text not as a sequence of individual characters, but rather as one line of text that is processed as a whole. A method for printed text recognition using this approach is described in (LeCun et al., 1998; Savchynskyy and Kamotskyy, 2006).

In our approach, the classifier is defined by a text structure (i.e. a grammar and a layout) and by a vector of parameters, representing the optimal appearance of individual characters. The method is based on a linear classifier. The classifier parameters are optimized according to a training set of examples using the structured support vector machine (SVM) learning method.

The next section describes how the text structure is modeled and sections 3 and 4 are focused on the learning and classification task. Finally we present experiments that were performed on car license plate and ADR/RID images.

2 TEXT STRUCTURE MODELLING

A digitized greyscale image I is a $H \times W$ matrix, elements of which are the intensity values of corresponding pixels. A *segment* of width $\omega \in \mathbb{N}$ of the image I is a submatrix of I formed by ω succesive columns. The *left border* λ of the segment is the index of the leftmost column of the segment (the lowest index). Each segment can be fully described by a pair (λ, ω) and also each pair $(\lambda, \omega) \in \mathbb{N}^2,\ \lambda + \omega \leq W + 1$ defines a segment in I. We will denote a segment of I with left border λ and width ω as $I[\lambda, \omega]$. The element in the i-th row and j-th column of I will be denoted I_{ij} and $I_{ij}[\lambda, \omega]$ for the segment $I[\lambda, \omega]$.

It is assumed that the input image depicts the text in a horizontal position and that the top and bottom edge of the image coincides with the top and bottom of the text. Neither the left-to-right position nor the width of the text is known, these are considered as unknown parameters, which makes it possible to cope with an imprecise detection of the left and right text border.

The text structure is described by a geometric model. A model μ is given by a sequence of segments the text contains. Each segment is described by its left border λ (i.e. its position), width ω and a type identifier. The *type* identifier is a subset of alphabet A containing all characters possibly appearing in a given segment. Thus the model has the form of a $3 \times N_\mu$ table, where N_μ is the number of segments. Figure 2 shows a typical model of a license plate text. The spaces between characters are modelled by a sequence of special space segments of width equal to one. On the other hand, the rest of the image that is not covered by the model is omitted.

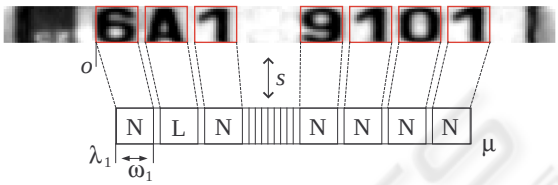

Figure 2: Typical structured text model. N stands for a type identifier denoting numbers, L means letters and empty narrow segments contain only space characters.

Because the width of the text in the image is unknown and the width of the model is fixed, it is necessary to find the ratio between the two. This ratio is called *scale*. The next unknown parameter is the left-to-right position of the text which is described by the index of its leftmost column called *offset*.

The combination of a model μ, scale s and offset o determines the geometry of the text and the model also defines all possible strings. The string will be denoted Σ and thus the complete description of a given image consists of four parameters – μ, s, o and Σ. We will also denote I^s the image I that has been resized by scale s to dimensions [1] $H \times \lceil W \cdot s \rceil$.

3 STRUCTURED SVM

Classification is a process that assigns a state from a given set of all possible states to an object, based on some observation made on the object. Classifier (or classification strategy) is a function $f : X \to Y$ that assigns to each observation $x \in X$ a state $y \in Y$. Next let us define a loss function $\Delta : Y \times Y \to \mathbb{R}$ that assigns to each pair $(y, f(x))$ a real number *loss* expressing the penalty for classifying x into $f(x)$, while the real state is y. We assume that $\Delta(y,y') = 0$ if $y = y'$ and $\Delta(y,y') > 0$ if $y \neq y'$.

The structured support vector machine learning method is based on finding an optimal classification strategy that minimizes the empirical risk defined as

$$R_{emp}(f) = \frac{1}{m} \sum_{k=1}^{m} \Delta(y_k, f(x_k)), \qquad (1)$$

[1] The $\lceil \cdot \rceil$ denotes the nearest integer towards infinity – ceiling.

and maximizes the margin supposing that there is a set of example data $\{(x_1,y_1),\ldots,(x_m,y_m)\}$ available (Vapnik, 1998). Here y_k denotes the true state of x_k.

To choose the optimal decision strategy, it is first necessary to determine the set of functions from which the optimal function should be chosen. Usually a set of functions is described by a function $F(w;x,y)$ dependent on some vector of parameters w. Then the decision strategy has the form of

$$\hat{y} = f(w;x) = \arg\max_{y\in Y} F(w;x,y) \tag{2}$$

and choosing the optimal strategy means choosing the optimal parameter vector w.

If a classifier is linear in the vector of its parameters, the optimal vector of parameters can be found using the structured support vector machine (SVM) learning algorithm. There are two main differences between classical and structured SVM. First, structured SVM allows for much more complicated output spaces than classical SVM, where the output space is merely a set of class labels. Second, arbitrary loss functions may be used, satisfying only previously mentioned conditions.

A classifier linear in the vector of its parameters w can be expressed as an inner product of the vector w and some vector function $\Psi(x,y)$ of the observation x and state y. This means that (2) takes the form of

$$\hat{y} = f(w;x) = \arg\max_{y\in Y} \langle w, \Psi(x,y)\rangle. \tag{3}$$

In the case described in this article, the state y is defined by a combination of four parameters – scale s, offset o, model μ and string Σ – introduced in section 2. Observation x corresponds to the input image I. Substituting these in (3) we can write

$$(\hat{s},\hat{\mu},\hat{o},\hat{\Sigma}) = \arg\max_{s,\mu,o,\Sigma} \langle w, \Psi(I,(s,o,\mu,\Sigma))\rangle. \tag{4}$$

The vector w represents prototypes of all characters a of the alphabet A. Prototypes $E(a)$ are images that all have the same height H as the input image. Vector w is created by placing these images column-wise after each other in a given order. The inner product $\langle w, \Psi(I,(s,o,\mu,\Sigma))\rangle$ expresses a similarity function of input image I resized by scale s and an image created from prototypes of characters in string Σ placed according to the model μ and offset o. We use the general form of the similarity function suggested in (Savchynskyy and Kamotskyy, 2006)

$$\langle w, \Psi(I,(s,o,\mu,\Sigma))\rangle = \sum_{i=1}^{N_\mu} \left(E(a_i) \odot I^s[o+\lambda_i,\omega_i]\right), \tag{5}$$

where $\Sigma = (a_1,\ldots,a_{N_\mu})$ is a string, $E(a_i)$ is the prototype of the character a_i and the $\odot$ operator denotes the similarity function between the prototype and the segment $I^s[o+\lambda_i,\omega_i]$. We use the cross correlation function for this purpose as described in (Franc and Hlaváč, 2006)

$$E(a_i) \odot I^s[o+\lambda_i,\omega_i] = \sum_{j=1}^{H}\sum_{k=1}^{\omega_i} E(a_i)_{jk} I^s_{jk}[o+\lambda_i,\omega_i], \tag{6}$$

The mapping function Ψ is thus defined implicitly by the equations (5) and (6).

The vector $\Psi(I,(s,o,\mu,\Sigma))$ can also be constructed explicitly by placing the segment $I^s[o+\lambda_i,\omega_i]$ to the vector Ψ in the same way as the prototype $E(a_i)$ is placed in the vector w (figure 3). The remaining elements of Ψ are set to zero so that these do not influence the value of (5).

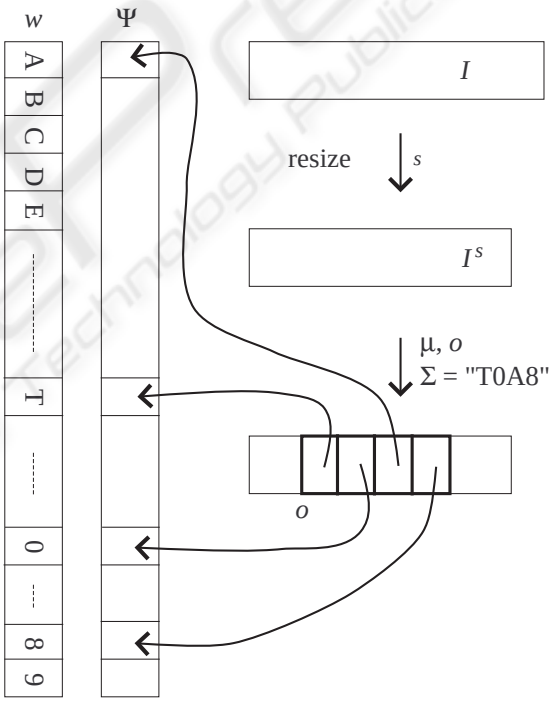

Figure 3: An example of construction of vector Ψ from image I. First, I is resized to I^s, second a model μ is placed on I^s on position o and finally segments are placed into Ψ on positions corresponding to characters in Σ.

Finding the optimal vector of parameters w in the sense of minimization of the empirical risk (1) and maximization of the margin is a QP optimization problem in the following form

$$\min_w \frac{1}{2}||w||^2 + \frac{C}{m}\sum_{k=1}^{m}\xi_k \tag{7}$$

such that

$$\forall k = 1,\ldots,m,\ \forall y \in Y :$$
$$\langle w, \Psi(x_k,y_k) - \Psi(x_k,y)\rangle \quad \geq \quad \Delta(y,y_k) - \xi_k, \quad (8)$$

where ξ_k are so called slack variables and C is a constant expressing the trade off between margin maximization and empirical risk minimization (Vapnik, 1998). Due to the large number of constraints (8), the QP task is performed iteratively. Most violated constraints are added to the working set in each iteration. Finding these constraints requires that there exists an algorithm for solving the so called *loss augmented classification task*

$$\hat{y} = \arg\max_{y \in Y} (\Delta(y,y_k) + \langle w, \Psi(x_k,y)\rangle). \quad (9)$$

The maximum in (9) is searched over all $y \in Y$. Since y is given by the parameters (s,μ,o,Σ), the geometric models are also used in the optimization (learning) process.

The correct segmentation of all images in the training sets is known and it is given by the states y_k. Thus each column of the training image can be labeled according to the character it depicts. The loss function $\Delta(y,y_k)$ was defined as the number of incorrectly labeled image columns for segmentation based y.

A general algorithm solving the problem (7) is described in (Tsochantaridis et al., 2005) and needs an external QP solver. A modified algorithm used in this work is described in detail in (Franc and Hlaváč, 2006).

4 CLASSIFICATION TASK EVALUATION

The recognition algorithm implements the maximization of (5) over all variables, i.e.

$$(\hat{s},\hat{\mu},\hat{o},\hat{\Sigma}) = \arg\max_{s,\mu,o,\Sigma} \langle w, \Psi(I,(s,o,\mu,\Sigma))\rangle \quad =$$
$$= \quad \arg\max_{s,\mu,o,\Sigma} \textstyle\sum_{i=1}^{N_\mu} (E(a_i) \odot I^s[o+\lambda_i,\ \omega_i]). \quad (10)$$

Since the model assumes that characters in different segments are independent of each other, the similarity function can be maximized within each segment separately.

$$(\hat{s},\hat{\mu},\hat{o},\hat{\Sigma}) =$$
$$= \arg\max_{s,\mu,o} \sum_{i=1}^{N_\mu} \max_{a_i} (E(a_i) \odot I^s[o+\lambda_i,\omega_i]). \quad (11)$$

The algorithm based on equation (11) is shown in figure 4.

Input:
Image I of height H and width W
A set of models $\mathcal{M}$
Prototypes $E(a)$ for all symbols a
Set of scales $\mathcal{S}$ and set of offsets $\mathcal{O}$
Output:
Scale $\hat{s}$, offset $\hat{o}$, model $\hat{\mu}$ and string $\hat{\Sigma} = (\hat{a_1},\ldots,\hat{a_n})$ maximizing the similarity function.

```
begin
    TOTALMAX := −∞
    forall s ∈ S do
        I^s = resize(I, s)
        forall μ ∈ M do
            forall o ∈ O do
                VALUE := 0
                Initialize array CHAR of length N_μ
                for i = 1 to N_μ do
                    MAXC := −∞
                    foreach a_i do
                        C := E(a_i) ⊙ I^s[o + λ_i, ω_i]
                        if C > MAXC then
                            MAXC := C
                            CHAR[i] := a_i
                        end
                    end
                    VALUE := VALUE + MAXC
                end
                if VALUE > TOTALMAX then
                    TOTALMAX := VALUE
                    μ̂ := μ   ô := o   ŝ := s
                    Σ̂ := CHAR
                end
            end
        end
    end
end
```

Figure 4: Basic algorithm for similarity function maximization.

5 EXPERIMENTS

In this paper we present experiments on three data sets. The first data set consists of car license plates from four European countries (Czech, Hungarian, Slovak and Polish). The second data set contains Saudi-Arabian license plates and the third set contains ADR/RID plates.

The first set consists of 2121 training images and 521 testing images. The input image size was 13x200 pixels. Eight models in total were used to describe the geometry and the syntax of the strings in the set. The recognized alphabet consists of 39 symbols. Although distinct text fonts appear in the set, just one prototype per character from the alphabet was used.

The second data set with Saudi-Arabian license plates consists of 627 training examples and 157 testing examples. Only one geometrical model with the

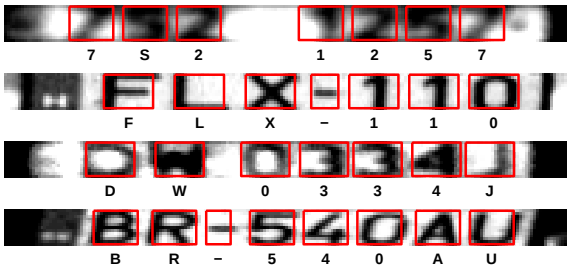

Figure 4: Four examples of input images from the first data set with recovered segmentation and recognized strings.

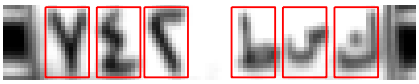

Figure 5: An example of Saudi-Arabian license plate image with recovered segmentation.

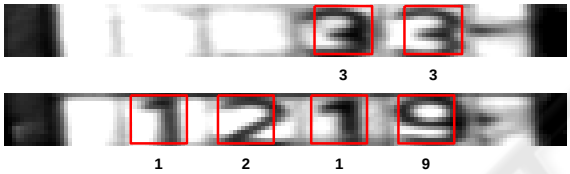

Figure 6: An example of a top and a bottom line from ADR plate with recovered segmentation and recognized strings.

alphabet of 27 symbols was used. The input image size was 24x100 pixels.

The third data set contains two-line ADR/RID plates. The set consists of 109 training and only 20 testing images. Each text line was recognized independently. The image resolution of the ipnut line was 13x140 pixels.

Several examples of input images and OCR results are shown in Figure 4, Figure 5 and Figure 6.

The total error rates achieved by the algorithm on the testing sets are given in Table 1, Table 2 and Table 3. Most of the errors are due to character misclassification. The segmentation error is typically low in this approach. If necessary, the total error can be reduced joining a nonlinear character classification module which cuts down the character misclassification error.

In general, the error rate depends on the quality of the input image sets and the complexity of the given recognition task (i.e. the number of all possible solutions). Unfortunately we did not find any public reference data set to enable the objective evaluation of the

Table 1: Error rates on data set consisting of Czech, Hungarian, Polish and Slovak license plates.

algorithm	total error	segmentation error	character misclsf
reference alg.	10.1%	3.3%	6.8%
proposed alg.	4.6%	0.95%	3.6%

Table 2: Error rates on Saudi-Arabian license plates.

algorithm	total error	segmentation error	character misclsf
reference alg.	18.1%	6.8 %	11.3%
proposed alg.	9.7 %	2.3 %	7.4%

Table 3: Error rates on ADR/RID plates.

algorithm	total error	segmentation error	character misclsf
reference alg.	5%	5%	0.0%
proposed alg.	0.0%	0.0%	0.0%

presented algorithm.

Therefore we took as a reference another algorithm described in (Franc and Hlaváč, 2006). This reference algorithm is also based on structured SVM, however it does not make use of any geometrical or syntax model.

6 CONCLUSIONS

In this article we proposed an OCR algorithm for structured texts that is based on exploiting the knowledge about their geometric and grammatical structure.

We introduced a formal description of a large variety of structured texts in terms of a geometric model. We also formulated the classification task in terms of maximizing a similarity function based on (Savchynskyy and Kamotskyy, 2006; Franc and Hlaváč, 2006) that compares the input image to an idealized one for all possible configurations. The idealized image consists of prototypes of individual characters. These prototypes are interpreted as parameters of the classifier that are to be determined by learning. We used the SVM method for structured classifiers described in (Franc and Hlaváč, 2006).

The described OCR method was tested in many experiments and currently was proved as a part of a commercial license plate recognition system. The algorithm fits especially for low quality images of strings with limited number of geometric and grammatical models.

ACKNOWLEDGEMENTS

This work has been sponsored by The Czech Ministry of Education project 1M0567 (M.U.) and by The Czech Science Foundation project 201/06/1821 (J.R.). The third author (V.F.) was supported by Marie Curie Intra-European Fellowship grant SCOLES MEIF-CT-2006-042107.

REFERENCES

Franc, V. and Hlaváč, V. (2006). A novel algorithm for learning support vector machines with structured output spaces. Research Report K333–22/06, CTU–CMP–2006–04, Department of Cybernetics, Faculty of Electrical Engineering Czech Technical University, Prague, Czech Republic.

Ko, M.-A. and Kim, Y.-M. (2003). License plate surveillance system using weighted template matching. In *Proceedings of the 32nd Applied Imagery Pattern Recognition Workshop (AIPR'03)*. IEEE Computer Society.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

Lee, H.-J., Chen, S.-Y., and Wang, S.-Z. (2004). Extraction and recognition of license plates of motorcycles and vehicles on highways. In *Proceedings of the 17th International Conference on Pattern Recognition (ICPR'04)*. IEEE Computer Society.

Rahman, C. A., Badawy, W., and Radmanesh, A. (2003). A real time vehicle's license plate recognition system. In *Proceedings of the IEEE Conference on Advanced Video and Signal Based Surveillance (AVSS'03)*. IEEE Computer Society.

Savchynskyy, B. and Kamotskyy, O. (2006). Character templates learning for textual image recognition as an example of learning in structural recognition. In *Proceedings of the Second International Conference on Document Image Analysis for Libraries (DIAL'06)*, pages 88–95. IEEE Computer Society.

Shapiro, V. and Gluhchev, G. (2004). Multinational license plate recognition system: Segmentation and classification. In *Proceedings of the 17th International Conference on Pattern Recognition*, volume 4, pages 352–355.

Tsochantaridis, I., Joachims, T., Hofmann, T., and Altun, Y. (2005). Large margin methods for structured and interdependent output variables. *Journal of Machine Learning Research*, 6:1453–1484.

Vapnik, V. (1998). *Statistical Learning Theory*. John Wiley & Sons, Inc.