# A classification architecture based on connected components for text detection in unconstrained environments

Luca Zini[1]
luca.zini@gmail.com

Augusto Destrero[1,2]
augusto.destrero@imavis.com

Francesca Odone[1]
odone@disi.unige.it

[1]DISI - Università degli studi di Genova - Via Dodecaneso 35 - I-16146 Genova, IT
[2]IMAVIS S.r.l. - Via Greto di Cornigliano 6R - I-16152 Genova, IT

## Abstract

*The paper presents a method for efficient text detection in unconstrained environments, based on image features derived from connected components and on a classification architecture implementing a focus of attention approach. The main application motivating the work is container code detection with the final goal of checking freight trains composition. Although the method is strongly influenced by the application experimental evidence speaks in favour of its generality: we present results on container codes, car plates images and on the benchmark dataset ICDAR.*

## 1. Introduction

Automatic text detection in unconstrained environments plays a central role in many computer vision applications as a first crucial step towards text recognition. The interest of the research community is testified by dedicated workshops and challenges — see for instance the ICDAR conferences. Focusing to the case of printed text different approaches may be devised, according to the amount of prior knowledge on the problem of interest. On one side of the spectrum we find well established applications, such as Automatic Licence Plate Recognition (ALPR), where a very small error rate has to be reached, but one can benefit from a deep knowledge on characters fonts and appearance. On the other side we find multi-purpose methods aiming at detecting as much text as possible in real world scenarios, regardless the size of the sign, color and fonts adopted, and the amount of clutter in the scene.

The work we propose here addresses a well defined problem that lies in-between the two aforementioned. We focus on containers code detection to the purpose of monitoring the entrance of freight trains in stations and to check the train composition. In this application domain, we do not have any prior knowledge neither on the amount of clutter of the scene, nor we make assumptions on the appearance of the container (that may change in color, size, amount of dirt) and on the position of the code on the container surface. At the same time, we have some knowledge on the shape of the code characters. In our work the latter feature will be exploited implicitly with an accurate choice of training data in an example-based approach. The major requirement of the application is to keep the computational cost low, since container detection is only the first step of a more complex monitoring system.

There have been very few attempts of addressing container code detection and recognition (see, for instance, [4, 5, 15]). For what concerns, more in general, printed character detection and recognition, an analysis of the available literature may be started from the proceedings of dedicated conferences [8]. Much attention is devoted to detection problem, since once a character has been properly localized, many consolidated pattern recognition techniques are available. The highlight from ICDAR benchmarks is [2] that proposes an Adaboost apprach based on a set of features that are not reported in the work (therefore the approach is not repetible).

The method we propose is based on segmenting the image in Connected Components (CC) and classifying them in text and not-text with a classification architecture based on a focus-of-attention strategy that limits the amount of computation. Notice that, up to know, no prior information available from the system (such has the relative distance between container and camera) has been applied. Thus the method is rather general and it has been applied, not only to a dataset of container codes images (obtaining an equal error rate – e.e.r.– below 2%) but also to car plates images (e.e.r = 1.2%). Real world images from the ICDAR DB have also been segmented with success.

## 2. The proposed method

Following a well established pipeline [16], the proposed method has been organized in three main stages: *(1)* image segmentation, *(2)* connected components description, *(3)* classification. As for image segmentation we adopt a simple yet effective local approach, the NiBlack method [9], while for what concerns connected components descriptions we rely on a selection of features that capture different appearance peculiarities. The final classification step is performed with an architecture of classifiers that considers a trade-off between accuracy and computational issues. The remainder of this section details the devised method.

### 2.1. Image segmentation

The first step of CC-based approaches is image segmentation used to extract connected components from the analysed image. In spite of the enormous amount of work dedicated to this topic, image segmentation can be still regarded as an open problem, due to objects complexity, artifacts due to illumination, noise caused by textures and low contrast regions. Here we rely on the fact that no matter how complex the scene is, there is usually a rather strong contrast between a foreground printed text and the background. To address our problem, with an eye on system requirements we focus on methods that are computationally efficient. Our current image segmentation module implements the NiBlack method that, adopting a local approach, is faster than other methods well established in the literature (e.g., fuzzy C-means [8], or mean-shift [3]). Also it allows us to capture local changes between foreground and background and it is less sensitive to non-uniform illumination. As a brief reminder of the Niblack method, we recall that a pixel is classified according to the following thresholding rule:

$$\text{Niblack}(i,j) = \begin{cases} 1 & \text{if} \quad I(i,j) > T_{+}(i,j) \\ -1 & \text{if} \quad I(i,j) < T_{-}(i,j) \\ 0 & \text{otherwise} \end{cases}$$

where $T_{\mp}(i,j) = \mu(i,j,W_m) \mp k\sigma(i,j,W_s)$ $k$ is a constant value and $W_m$ $W_s$ is the patch size where mean and variance are computed ($1/5$ of image size in our case). The major strength of the Niblack approach is its simplicity and the fact that the algorithm is not too sensitive to parameter choice. For these reasons a number of variants have been recently proposed — see for instance [16, 13].

### 2.2. CC descriptions

Once a set of connected components has been localized within the image, an appropriate way to describe their appearance has to be devised. This description should be evocative of the problem of interest, i.e., it should produce a good discrimination between text characters and other objects. Our work relies on descriptions of the single character (as opposed to other approaches that first group adjacent features and then describe such groups [8]).

To this purpose we use a set of 20 features, most of them available from the literature. The list of adopted features is reported in Table 1. The table contains an appropriate citations, when available. The other features may be described as follows:

- **Homogeneity:** $\frac{\sigma^2(CC)}{\sigma^2(BB)}$ (BB stands for Bounding Box). This description represents the degree of color (or grey level) homogeneity within the character region. A variant uses a wider region than BB to deal with $CC \sim BB$ (as in the case of the character *I*).

- **Direction variations:** the direction variations along the CC contour is computed. This measures the regularity of the character.

- **Correlation:** $\sum_{(r,c)\in CC} I(r,c) \neq I(r-1,c)$ and $\sum_{(r,c)\in CC} I(r,c) \neq I(r,c-1)$. The two values are combined in different ways giving raise to 3 versions: (1): average value; (2): min value; (3): comparison between adjacent lines on a 8-connected component. This description evaluates the degree of correlation between a scanline (row or column) and the adjacent one.

- **Stroke width:** this description computes the average stroke and its standard deviation; average is normalized with the connected component height. It captures the stroke stability typical of printed characters.

Also, we consider a measure of contour roughness [16] that implements a morphological closure instead than an opening. This makes the measure more stable when dealing with small characters.

### 2.3. Classification architecture

The classification architecture that we consider is based on Regularized Least Squares (RLS) — other regularized algorithms could be easily adopted, with some computational gain on the training phase [6]. In the classification phase we train a classifier on a training set of positive (text connected components) and negative (non text) examples described according to the previously described CC-features. Since the CC-features are heterogeneous the issue on how to combine such features needs to be addressed.

A first approach is to build a global feature vector after all features are normalized to a common range of values (e.g., [0,1]). The major disadvantage with this approach is computational: since we deal with an object detection problem we aim at minimizing the number of evaluations for each analysed image region.

| **feature** | **reference** |
|---|---|
| Aspect ratio | [16] |
| Occupy ratio | [16] |
| Edge contrast | [16] |
| Homogeneity | see text |
| Homogeneity (variant) | see text |
| Edge Symmetry | [7] |
| Run length | [1] |
| Direction variations | see text |
| Contour roughness (open) | [16] |
| Contour roughness (close) | see text |
| CC Holes | [16] |
| Perimeter length | [14] |
| Correlation ver. 1, 2, 3 | see text |
| Stroke width | see text |
| Zernike moments | [14] |
| Generalized Fourier Descriptor | [14] |
| Normalized central moments | [14] |
| Hu moments | [14] |

Table 1. List of the 20 descriptions adopted. The reader is referred to the cited works or to the text.

On this respect a possible attractive alternative is to adopt a *focus of attention* approach that allows for a fast elimination of negative data. The idea of such architectures has been widely adopted for object detection problems, to deal with the strong unbalance between positives and negatives in a given image. This issue has been debated, for instance, in [11] where a *cascade of classifiers* is implemented. The underlying idea is that, with a careful composition of independent weak classifiers, one can achieve very good overall performances. Each classifier is trained so that it will not be likely to miss positive occurrences, to the price of increasing the number of false positives. The output $c(x)$ of each classifier is modified as follows:

$$g(x) = \begin{cases} 1 & c(x) \geq \tau \\ -1 & \text{otherwise} \end{cases} \qquad (1)$$

where the threshold $\tau$ is set so to obtain a given hit rate. The overall performance of the independent classifiers is obtained as $H = \prod_{i=1}^{K} h_i$ and $F = \prod_{i=1}^{K} f_i$ where $h_i$ is the hit rate and $f_i$ is the false positive rate for each weak classifier $i$. Notice that if we set the minimum hit rate to $99.5\%$ and the maximun false positive rate to $50\%$, these modest targets allow us to achieve good performances with the global classifier, for instance, assuming a cascading of 10 weak classifiers, we will get $H = 0.995^{10} \sim 0.9$ and $F = 0.5^{10} \sim 3 \times 10^{-5}$.

The cascade is built by first splitting the available features in different groups. Each group is used to train a classifier and to tune it to the required performances. At run time, each test element is evaluated with the first classifier: if the classifier returns a negative output, the element is discarded, otherwise it is used as an input for the second classifier. A positive output is associated to the test element only if it passes all classifiers. The required precision may be estimated with a cross-validation procedure, selecting the threshold that guarantees the best trade-off between a high hit rate on each layer and a good overall false positives rate.

For what concerns the choice on how to combine features in a cascade, for computational reasons, we choose small feature vectors per each layer. The simplest choice is to associate one feature to each layer. In this case we need to find an ordering for the features, from the most to the least informative for the problem at hand — this can be done by means of feature selection. This choice leads to a *single-feature cascade*. We also combine features considering *at most* two features at the time. Again, to choose significant feature pairs we apply a feature selection method on all possible feature pairs and single features. This choice leads to a *at-most-2-features cascade*.

In this paper we apply a recently proposed feature selection technique: Kernel Class Separability [12]. The method performs feature selection extending a rather simple linear feature selection method (class separability) to deal with non linearly separable classes, by means of an appropriate mapping in a kernel space. The criterion is based on $\mathcal{J}_0 = \frac{tr(S_B)}{tr(S_W)}$ where the within-class scatter matrix $S_W$ and the between-class scatter matrix $S_B$ are computed in the feature space induced by a kernel $K$. To maintain the numerical stability of the criterion, the authors use a lower bound for $\mathcal{J}_0$: $\mathcal{J}_l = tr(S_B)$. Then, instead than using it straightforwardly they apply a *maximal class separability* over the kernel parameter set $\theta$, as a way to perform directly parameter tuning: $J(\theta^*) = \max_{\theta \in \Theta[J(\theta)]}$. If the selected kernel function has a first and second-degree derivative (as in the case of the Gaussian RBF kernel adopted in our work), then a gradient-based optimization technique can be used to solve the maximization problem.

## 3. Experimental analysis

### 3.1. The datasets

In the experimental evaluation phase we rely on two image datasets: a car plate image set (121 images, 512 $\times$ 256) and a container code image set (100 images, 1200 $\times$ 1600), all acquired under natural daylight illumination and variable weather conditions. All images have been segmented with the NiBlack algorithm and the obtained connected components have been manually labelled in text and non text elements. A first dataset, 500+500 training data and 300+20460 test data, is built from the car plates images (the reason for such an unbalanced dataset is due to the intrinsic nature of object detection: the number of positive occurrences are much smaller than negative ones). Most of the method assessment has been carried out on this dataset.

| Feature | Linear RLS | Gaussian RLS |
|---|---|---|
| Aspect ratio | 0.87 | 0.89 |
| Occupy ratio | 0.51 | 0.64 |
| Edge contrast | 0.75 | 0.82 |
| Homogeneity | 0.45 | 0.82 |
| Homogeneity (variant) | 0.44 | 0.78 |
| Edge symmetry | 0.54 | 0.60 |
| Run length | 0.73 | 0.89 |
| Direction variations | 0.51 | 0.71 |
| Contour roughness (open) | 0.60 | 0.62 |
| Contour roughness (close) | 0.63 | 0.74 |
| CC Holes | 0.55 | 0.62 |
| Perimeter length | 0.70 | 0.72 |
| Correlation 1 | 0.75 | 0.83 |
| Correlation 2 | 0.55 | 0.55 |
| Correlation 3 | 0.60 | 0.69 |
| Stroke width | 0.75 | 0.85 |
| Zernike moments | 0.90 | 0.96 |
| Generalized Fourier Descriptor | 0.89 | 0.93 |
| Normalized central moments | 0.85 | 0.91 |
| Hu moments | 0.61 | 0.85 |

Table 2. Cross-validation performances obtained with single feature classifiers. The results show that a Gaussian kernel is more appropriate for the problem of interest. They also show a high degree of variability on the performance of the various features.

To deal with the text localization problem on the containers dataset we built a training set combining container and car plates data, while the test set is made of CC from containers images. The mixed dataset is composed of 500+500 training entries and 970+40000 test entries. To assess the whole object (text) detection process we use a set of containers images and a separate test set of unconstrained images used as a benchmark in the literature: the ICDAR dataset [8]. We conclude with a note on segmentation performances: we achieve 85% correct characters segmentation on car images and 94% on container images. Lower resolution images are more complex, as adjacent characters tend to merge.

## 3.2. Classification experiments

The classifiers implement a RLS approach. As for the choice of the kernel function we are currently adopting a Gaussian kernel. A preliminary analysis of the discrimination power of single features (Table 2) allowed us to obtain a simple comparison between linear and Gaussian classifiers, with a clear superiority of the latter. We start off with the car plates dataset and first evaluate the performance of the unique classifier trained on a feature vector based on all features. Then, before we move on to build the classifiers cascade a Spearman correlation test was run among all features pairs, computed on the positive examples. The results obtained are always below 0.3, showing a small correlation between the various features adopted and thus justifying the use of a cascading architecture. We built a single-

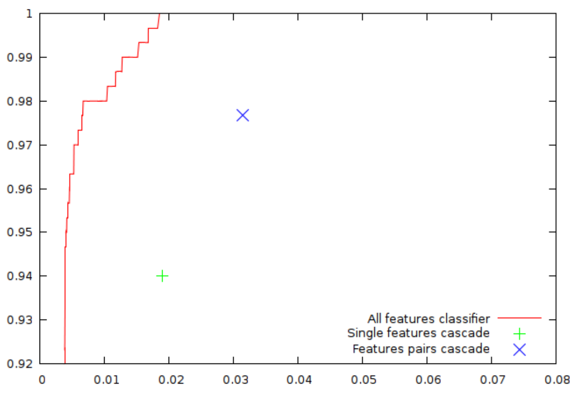

Figure 1. Comparison between the ROC curce of the unique classifier and performances of the two types of cascades evaluated (see text).

features cascade, selecting the most appropriate minimum hit rate for each classifier according to the performances of the single classifiers (evaluated with a Leave-One-Out (LOO) cross validation). The selected minumum hit rate is 99.5%. With a cascade based on single feature classifiers we obtain a hit rate of 94% with 1.8% false positives.

Finally, we run the feature selection procedure to build a cascade of at-most-2-features classifiers. The procedure produces 6 pairs : (homogeneity (variant), correlation 2), (perimeter length, normalized central moments), (contour roughness (close), correlation 1), (stroke width, correlation 3), (occupy ratio, Hu moments), (contour roughness (open), CC holes). The other features form single-features layers. Again with LOO-cross validation we select the minimum hit rate to be reached by each classifier. The results obtained are a 98% hit rate with a 3.1% false positives rate.

Figure 1 reports a Receiver Operating Characteristic (ROC) curve of the performance obtained with the unique classifier. The curve is built varying the threshold $\tau$ applied to the output of the RLS. It also shows the results obtained with the two types of cascade architectures. The at-most-2-features cascade performs better than the single feature cascade. Among the three choices, the unique classifier performs best, suggesting that the non-linear combination of features due to the kernel produces a richer and more informative description. At the same time, the adoption of a cascading mechanism makes the detection process on one image 8 times faster. Then, if computational issues are crucial we may adopt at-most-2-features cascade. Figs. 2 and 3 show segmentation results at different layers of the cascade. Correct text detection is above 90%, most errors are due to segmentation. The computational gain is due to the fact that image regions not painted in green will be discarded from further computations.

An alternative choice to boost correct detection perfor-

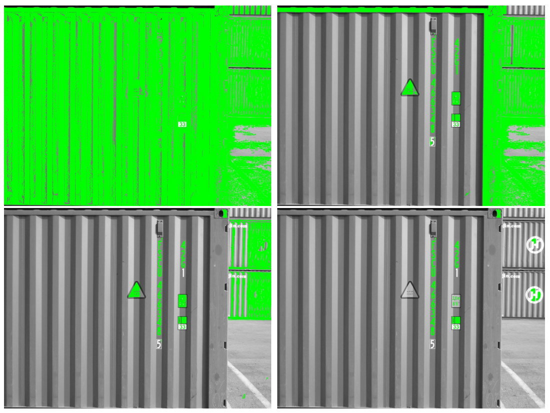

Figure 2. Detection results at different layers of the cascade. Detected text is painted in green. The final layer contains 7 false positives over 449 and 1 false negative over 22 correctly segmented characters.

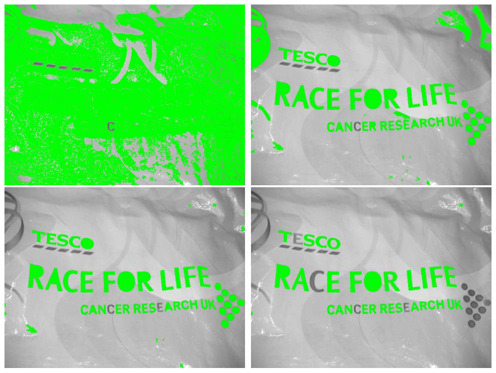

Figure 3. Detection results (ICDAR db) at different layers of the cascade. Detected text is painted in green. The final layer contains 4 false positives over 314 and 2 false negative over 30 correctly segmented characters.

mance while keeping computational cost low is to adopt a combined solution between cascades and monolithic classifiers considers, first, a few fast layers of a cascade, to discard "easy" negative examples (that are the majority), then a monolithic classifier trained on remaining features and equipped with a non-linear kernel. We implement a 2 layers cascade with the features aspect ratio and edge strength (the first two layers in the at-most-2-features cascade). Then, for those CC that pass the 2 layers test a unique classifier of the remaining features is applied. This combination allows us to reduce the amount of computation (only 20-30% of CC are evaluated with the unique classifier, making text detection four times faster). This combined procedure leads to a good compromise between performance (reported in Fig. 4) and speed (processing is 4 times faster).

The results are confirmed on the containers dataset. Figure 5 reports a comparison between the unique classifier, the

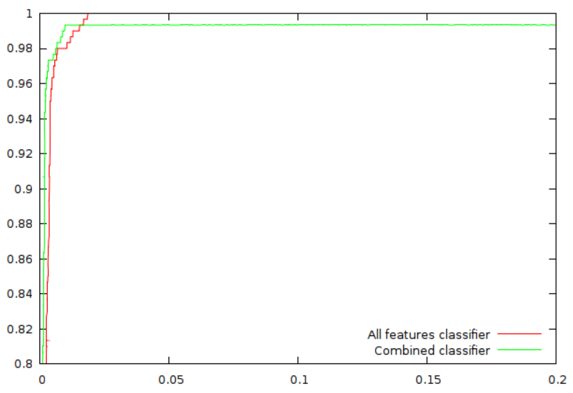

Figure 4. Comparison between the unique classifier ROC curve and the one of the combined solution (2 layers + monolithic classifier).The decrease in performance is very limited while processing is 4 times faster

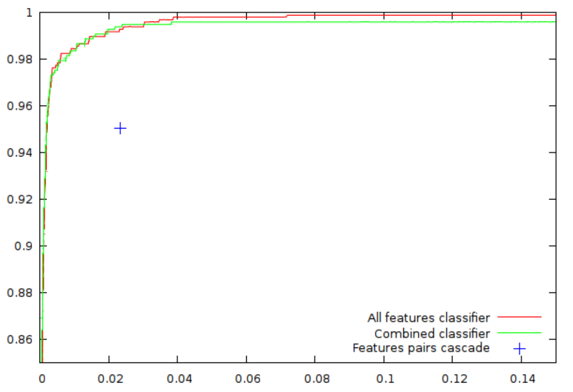

Figure 5. Compoarison on the containers dataset. Again the combined solution appears to be a very good compromise between speed and correct classification rates.

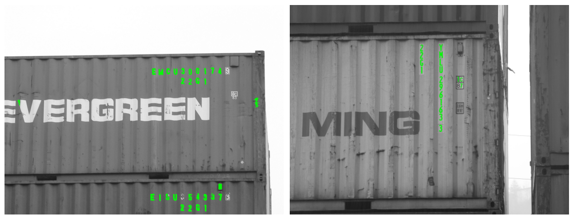

Figure 6. Text detection on low quality container images. All containers codes are correcly detected. Big characters are not detected since the focus of the application is on container codes and image segmentation parameters were tuned on their size.

combinination of a 2 layers cascade followed by the unique classifier, and the at-most-2-features cascade.

We conclude this section remarking that one of the main challenges of localizing container code is due to acquisition conditions that are often poor. This can be either due to

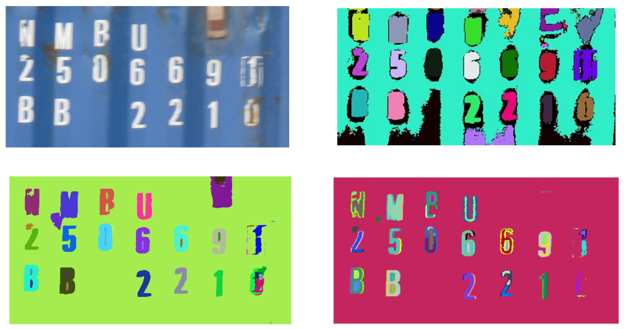

Figure 7. Noise and blur (typical of container codes images) strongly affects segmentation results. From top left (clockwise): original image, niblack, mean-shift, c-means.

weather conditions, dirt on the container, low contrast between text and background, distance between camera and container. Figure 6 reports correct text detection, obtained with the combined architecture, on very difficult images acquired from a distance in bad weather conditions.

## 4. Discussion

The paper reported a CC approach to text detection that implements a classification architecture based on focus of attention. We achieve satisfactory detection performances with a limited computational cost. The devised method is quite general (it has been applied with success to car plates, container codes, and to a generic benchmark dataset of printed characters) even though the main application motivating our work is container code localization.

An inspection on the weaknesses of our approach shows how the image segmentation module is very sensitive to image noise and blur: most misses are caused by the failure of this pre-processing stage. Figure 7 shows a low quality container code image and makes apparent the weakness of NiBlack (top right). It also shows the difficulty of the data we are dealing with, since other popular methods obtain unsatisfactory results — mean shift (bottom right) over-segments the characters, while some characters are missed with c-means (bottom left). Future work will be devoted to devise an alternative fast image segmentation method, we are currently exploring the possible use of [10].

**Acknowledgements** This work has been partly supported by the EU Project n. 222199 VIT "Vision for Innovative Transport" (Capacities - Research for SMEs).

## References

[1] Y.K. Chan and C.C. Chang. Image matching using run-length feature. *Patt. Recogn. Letters*, 22(5), 2001. 3

[2] X. Chen and A.L. Yuille. AdaBoost Learning for Detecting and Reading Text in City Scenes. *Proc. of IEEE Conference CVPR*, 2004. 1

[3] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on pattern analysis and machine intelligence*, 24(5):603–619, 2002. 2

[4] Salvador-Igual Ismael, Andreu-Garcma Gabriela, and Pirez-Jiminez Alberto-J. Preprocesing and recognition of characters in containers codes. In *International Conference on Pattern Recognition*, pages 1001–1005, 2002. 1

[5] Kwang-Baek Kim. Recognition of identifiers from shipping container images using fuzzy binarization and enhanced fuzzy neural network. In *Proceedings of the 2nd Int. Conf. on Fuzzy Systems and Knowledge Discovery*, volume Lecture Notes in Computer Science, pages 761–771, 2005. 1

[6] Lo Gerfo L., Rosasco L., Odone F., De Vito E., and Verri A. Spectral algorithms for supervised learning. *Neural Computation*, 7:1873–1897, 2008. 2

[7] XU Li, QI Fei-hu, J. Ren-jie, ZHU Kai-hua, and WU Guo-rong. A Novel Method for Character Segmentation in Natural Scenes. *Journal Of Shanghai Jiaotong University*, 11(4):484–489, 2006. 3

[8] S.M. Lucas, A. Panaretos, L. Sosa, A. Tang, S. Wong, R. Young, K. Ashida, H. Nagai, M. Okamoto, H. Yamamoto, et al. ICDAR 2003 robust reading competitions: entries, results, and future directions. *Int. J. on Document Analysis and Recognition*, 7(2):105–122, 2005. 1, 2, 4

[9] W. Niblack. *An introduction to digital image processing*. Strandberg Pub. Company, Denmark, 1985. 2

[10] R. Nock and F. Nielsen. Statistical region merging. *IEEE Trans. on PAMI*, 26(11), 2004. 6

[11] P. Viola and M. Jones. Robust real-time object detection. *IJCV*, 57(2):137–154, 2002. 3

[12] L. Wang. Feature selection with kernel class separability. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(9):1534–1546, 2008. 3

[13] C. Wolf and J.M. Jolion. Extraction and recognition of artificial text in multimedia documents. *Pattern Analysis & Applications*, 6(4):309–326, 2004. 2

[14] D. Zhang and G. Lu. Review of shape representation and description techniques. *Patt. Rec.*, 37(1):1–19, 2004. 3

[15] He Zhiwei, Liu Jilin, Ma Hongqing, and Li Peihong. A new localization method for container autorecognition system. In *IEEE Int. Conf. Neur. Net. and Sign. Proc.*, 2003. 1

[16] K. Zhu, F. Qi, R. Jiang, and L. Xu. Automatic character detection and segmentation in natural scene images. *Journal of Zhejiang University-Science A*, 8(1):63–71, 2007. 2, 3