Racing Bib Number Recognition

Idan Ben-Ami
idan.benami@gmail.com
Tali Basha
talib@eng.tau.ac.il
Shai Avidan
avidan@eng.tau.ac.il

School of Electrical Engineering,
Tel Aviv University,
Tel Aviv 69978, Israel

Abstract

We propose an automatic system for racing bib number (RBN) recognition in natural image collections covering running races such as marathons. An RBN is typically a piece of durable paper or cardboard bearing a number as well as the event/sponsor logo. The RBN, usually pinned onto the competitor's shirt, is used to identify the competitor among thousands of others during the race. Our system receives a set of natural images taken in running sport events and outputs the participants' RBNs. Today, RBN identification is often done manually, a process made difficult by the sheer number of available photos.

This specific application can be studied in the wider context of detecting and recognizing text in natural images of unstructured scenes. Existing methods that fall into this category fail to reliably recognize RBNs, due to the large variability in their appearance, size, and the deformations they undergo. By using the knowledge that the RBN is located on a person's body, our dedicated system overcomes these challenges and can be applied without any adjustments to images of various running races taken by professional as well as amateur photographers. First, we use a face detector to generate hypotheses regarding the RBN location and scale. We then adapt the stroke width transform (SWT) to detect the location of the tag, which is then processed and fed to a standard optical character recognition (OCR) engine. We evaluate the contributions of each component of our system, and compare its performance to state-of-the-art text detection methods, as well as to a commercially available, state-of-the-art license plate recognition (LPR) system, on three newly collected datasets.

1 Introduction

Running races, such as marathons, are broadly covered by professional as well as amateur photographers. This leads to a constantly growing number of photos covering a race, making the process of identifying a particular runner in such datasets difficult. Today, such identification is often done manually. In running races, each competitor has an identification number, called the Racing Bib Number (RBN), used to identify that competitor during the race. RBNs are usually printed on a paper or cardboard tag and pined onto the competitor's T-shirt during the race. We introduce an automatic system that receives a set of natural images taken in running sports events and outputs the participants' RBN.

Tag number recognition is commonly used in various traffic and security applications, such as parking, border control, and analysis of traffic flow. Many license plate recognition

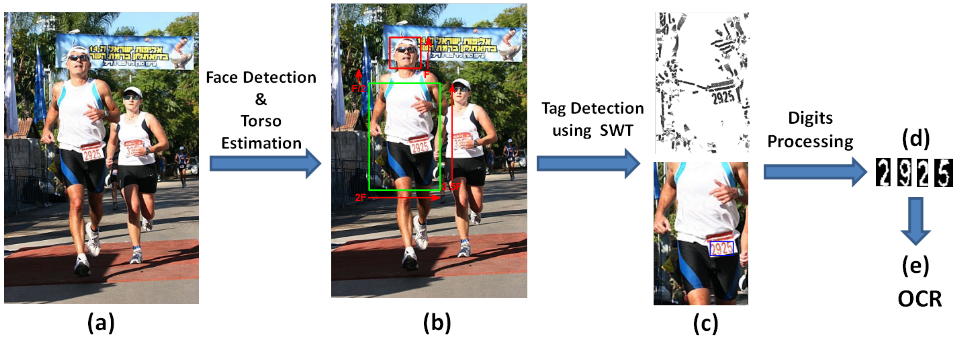

Figure 1: **Method Outline: (a)** the input image; **(b)** face detection results in red; the hypothesis estimated region of the RBN in green; **(c)** the stroke-width map of the hypothesis region (top) and the detected tag (bottom); **(d)** the detected tag after processing is fed to the OCR **(e)**.

(LPR) methods have been developed to automatically recognize the license plates on vehicles from still images and videos. A comprehensive survey of research on LPR is given in [1]. Despite the rapid development of commercial LPR systems, most of the LPR methods work under restricted conditions, such as fixed illumination, one vehicle per image, specific location in the image, specific font or color, and non-complex background. However, when considering natural sport images, none of these restrictions is valid. In particular, the tag numbers may be different in size, font and color. More importantly, while the license plate is a well-defined rigid rectangular shape, the RBN is wrapped on the runner's body. Thus, one has to deal with non-rigid deformation of the tag, which makes the problem of detecting and recognizing the number more challenging.

The problem of RBN recognition can also be investigated within the larger context of text in the "wild," where the goal is detect, and recognize, text in natural images of unconstrained scenes. This problem is gaining popularity with the rise of camera-equipped mobile phones. A number of recent studies have considered the problem of text detection [2, 3, 4, 6, 8, 9, 11, 12].

The stroke width transform (SWT) [4] relies on the fact that text appears as strokes of constant width. The problem is thus reduced to that of finding regions with constant width strokes. These regions can then be fed to an OCR engine. Another approach, recently introduced by Wang *et al*.[12], proposes an end-to-end text recognition system. The system assumes a fixed lexicon of words to be detected and relies on techniques from the field of general object recognition to detect those words. In particular, sliding-window classifiers (character detectors) are used to detect potential locations of characters in the image. The classifiers are trained in a prior stage using a set of synthetic images of characters with different fonts, Gaussian noise, and affine deformation. Both of these methods were shown to produce impressive results on natural images. However, directly applying them on our data is insufficient for reliably detecting the RBNs (as demonstrated in our experiments). The RBNs usually cover only a small portion of the image and are surrounded by complex backgrounds. Moreover, the images often contain irrelevant text such as sponsor billboards, signs, or text printed on people's clothes. Therefore, text detection methods are expected to be inefficient and to produce many false detections.

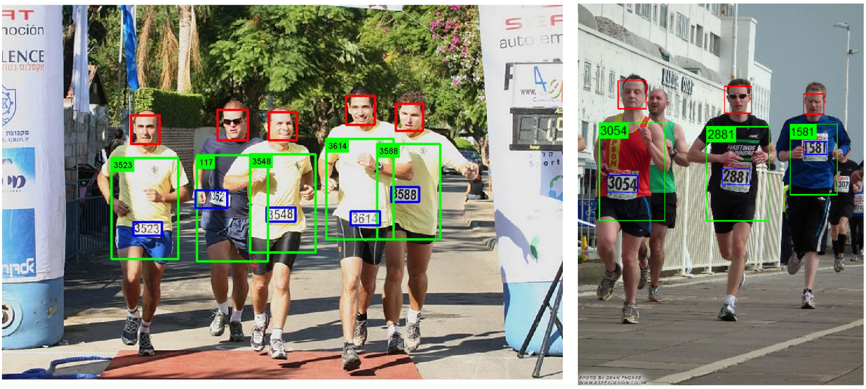

Figure 2: Detection of multiple RBNs per image.

In this paper, we propose a method specifically designed for RBN recognition (see outline in Fig. 1). Our method can be applied without any adjustments to images taken at various running races by different photographers. We show that by using prior information - the spatial relation between a person's face and the tag he/she wears, we obtain an effective RBNR system that outperforms text detection methods and state of-the-art commercial LPR software.

2 Method

The input to our method is a collection of images covering a running race. The images are generally different in size, resolution, and viewpoint, and may be taken using different cameras. Each image is assumed to capture one or more participants (e.g., Fig. 2). The RBN tags are allowed to have any color, font and scale. The only assumption is that the RBN tag is located on the front torso of the participant.

A straightforward approach for RBN recognition is to apply existing methods for locating text, e.g., SWT, on the entire image, and then use OCR on the detected regions. This approach is insufficient for reliable recognition for several reasons: the image background is complex – typically urban scenes with vehicles and other humans. Moreover, the RBNs are often surrounded by irrelevant text such as sponsor billboards, signs, or text printed on people's clothes. In addition, there is a large variability in the RBNs' appearance, size and location, and the RBNs usually cover only a small portion of the image (see Fig. 3). Therefore, text detection methods are expected to be inefficient and to produce many false and miss detections. This is demonstrated in Fig. 4(a)-(c), showing the results of applying SWT on the entire image. In addition, applying off-the-shelf OCR on the detected tags is expected to fail due the unrestricted properties of the tag. We next describe how our method overcomes these challenges and detail each its steps (see outline in Fig .1).

2.1 Face Detection

In the first step of our method, our goal is to reduce the RBN search area per image by using the knowledge that the RBN is located on a person's body. This is done by detecting faces,

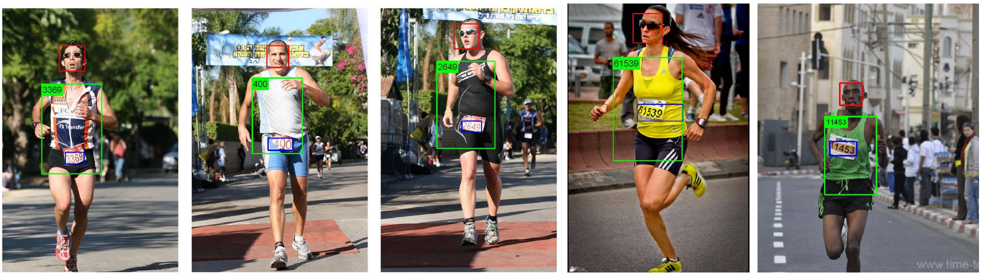

Figure 3: Examples of RBN detection and recognition: the detected face, the bounding box of the hypothesis tag region, and the detected tag are marked over the images; the final recognition result is shown on the left-hand upper corner of each bounding box.

and using their location and scale to generate a hypothesis about the locations of the RBN (see Fig. 1b). To avoid misses, we generate a very tolerant bounding box around the torso of the runner. In particular, the bounding box dimensions are: $3 \cdot \textit{(face height)} \times \frac{7}{3} \cdot \textit{(face width)}$, and it is located $0.5 \cdot \textit{(face height)}$ below the face. The estimated torso area is then used to guide the SWT search, as we describe next. Note that this approach allows us to deal with an unknown number of participants in the image.

2.2 Stroke Width Transform

We wish to detect the RBN location in each of the hypothesis regions detected as described above. To this end, we adopt the Stroke Width Transform (SWT) [4]. The SWT is a local image operator that computes the likely stroke width for every pixel in the image. Epshtein *et al*. [4] showed that the SWT can effectively be used to detect text regions in natural images by grouping components with stable stroke width, color and size. Since the SWT is mainly based on edges, its accuracy is limited when considering blurred text, small characters (1-2 pixels of width), connected characters or illumination changes inside the text region. In such cases, which are quite common in RBN images, the RBN cannot be detected. To improve the accuracy of the SWT we combine it with the following enhancements:

- A maximal allowed stroke width is extracted using the face scale, and is used to limit the digit candidate scales.
- Low resolution tags (where the text is expected to be 1-2 pixels width) are scaled up to allow standard computations such as edge extraction.

Since we significantly limit the search area, the SWT is applied only on a small portion of each image (the bounding box around the torso). In our datasets, the average portion of the search area per image is 15% of the total image size. Combined with these modifications, the SWT can provide very good RBN detection performance with relatively high precision (low false positive rate) and efficient processing.

2.3 Digit Processing

At this point, each participant is assigned one or more RBN tag candidates. Applying OCR directly on the detected tags is prone to fail due to the non-rigidity of the tag, changes in

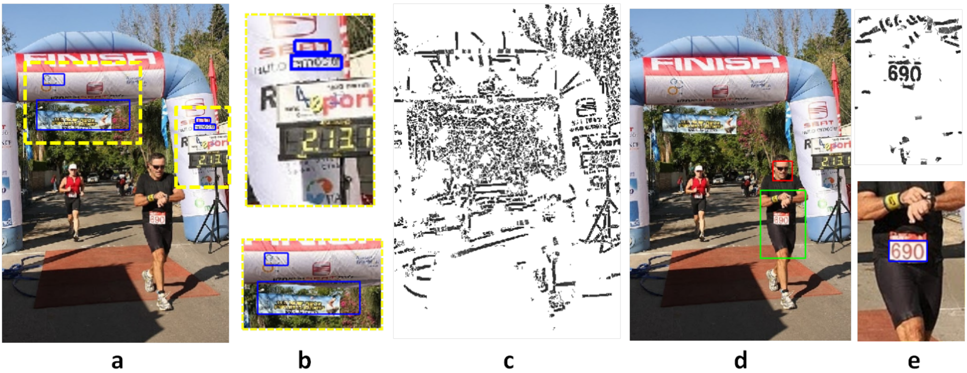

Figure 4: **(a)** The result of applying SWT on the entire image; the detected text regions are marked in blue inside the yellow dashed regions; **(b)** zoom-in of the yellow dashed regions; **(c)** the stroke width map computed on the entire image; **(d)** the detected face in red, and the hypothesis tag region in green; **(e)** the result of applying SWT with our enhancements (Sec.2.2) on the hypothesis region; the stroke width map (top), and the detected tag (bottom).

illumination along the tag, noise, and detection errors. Therefore, in essence, our method processes and converts each tag candidate to a binary clean image, as described below.

Segmentation and Binary Conversion: To deal with the non-rigidity of the tag and the illumination changes along it, we segment each tag into its constituent components (characters), and process each of the tag components separately. The initial segmentation of each tag is obtained in the SWT stage, but it may be inaccurate. In many cases the stroke width of a character can be incorrect or missing in some parts, for example in the character's corners. In some cases the stroke width of one character can be segmented into two or more fractures. In order to achieve a better quality binary image of the character, we continue as follows. The tag is initially segmented according to the approximate locations of its components, computed in the SWT stage (see Fig. 5(b)). Each segment is then converted to a binary segment using the average RGB color of the estimated component's pixels (see Fig. 5(c)). In particular, we threshold the L1 distance between the estimated average RGB color and the color of each pixel in the segment. The resulting binary tag usually contains a sufficient description of the digit, unaffected by the illumination gradient between characters, character color, or background color. However, the binary tag may also contain wrong components and noise such as text printed on the RBN tag, folds in the runner's garment that hide part of the RBN and folds in the RBN tag. We filter out such components according to their size, location and orientation (see Fig. 5)(d)).

Rotation: In the final step, we align each character according to its estimated orientation (see Fig. 5)(e)). The orientation of each character is computed using the ellipse which has the same second moment as the character region in the binary image. The orientation is the angle between the horizontal and the major axes of the ellipse. Note that we approximated the deformation of the tag by considering 2D rotations of each character. A more sophisticated modeling of the deformation is expected to improve the results.

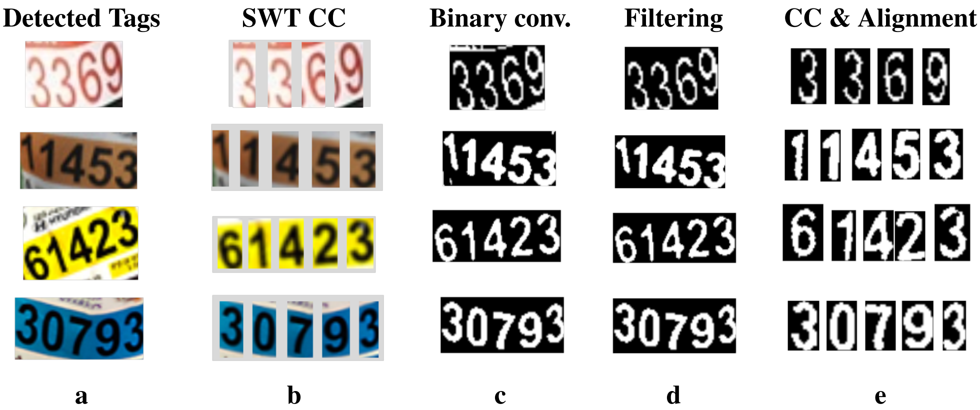

Figure 5: **Digit processing:** **(a)** The detected tags; **(b)** initial character separation according to SWT connected components (CC) analysis; **(c)** binary conversion of each character. **(d)** connected components analysis and filtering; **(e)** final separation and alignment.

2.4 OCR

Character recognition is carried out using the Tesseract OCR engine [10]. The aligned binary image of each digit is fed to the OCR and analyzed in order to find the best matched prototype digit from 0-9. No assumptions are made regarding the OCR training data. In fact, we use the Standard English training set which is included in the Tesseract package. Tesseract generates a confidence level for the character recognition, which is minus the normalized distance from the character prototype. We use the OCR confidence level to remove erroneous components from the RBN. That is, we remove components with significantly low confidence with respect to others in the RBN (less than 0.7 of the median confidence value of the RBN).

2.5 RBN Tag Selection

So far we have dealt with a single detected RBN tag per hypothesis region (i.e., the torso bounding box). However, it is quite common that multiple RBN tag candidates are detected. Several rules are used in order to filter out wrong candidates. These rules are based on two assumptions: (1) the size of the RBN is larger than quarter of the face scale in each dimension, and smaller than 85% of the torso bounding box (see Fig. 6); and (2) if there is more than one RBN candidates left at this stage, the decision as to which one to choose is made on the basis of the average confidence values of the digits.

	Resolution	# of Images	# of RBNs
Set #1	342x512 - 480x720	92	100
Set #2	800x530 - 850x1260	67	75
Set #3	768x1024	58	111

Table 1: **Datasets**: Image resolution, the number of images and the number of ground-truth RBNs in each dataset.

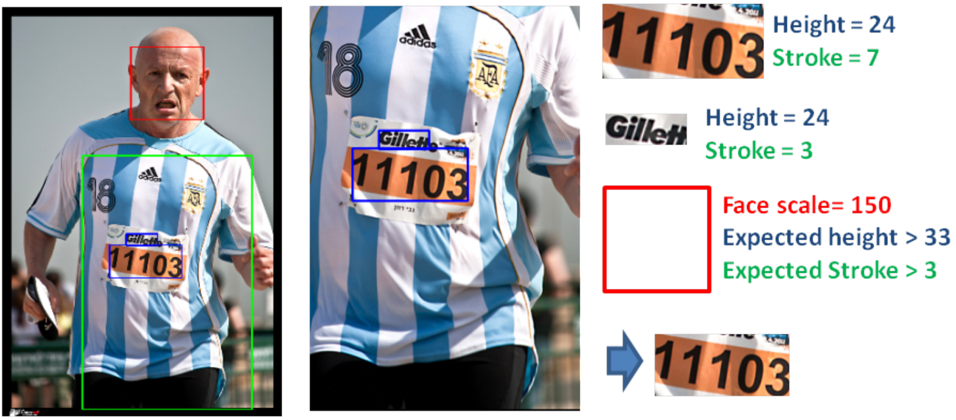

Figure 6: Tag Selection: **(a)** The tag is searched in the green bounding box; **(b)** two tags are detected – marked in blue. **(c)** the face scale is used to limit the size and stroke of the tag, and filter out wrong candidates.

3 Results

To test our method, we collected images from running races found on the Web. Our database includes 217 color images divided into three sets, each taken from a different race. Table 1 summarizes the description of the sets. The tag dimensions vary between 13x28–120x182 pixels while digit stroke widths vary from 13 pixels to as few as 2 pixels in the smallest tags. To verify and compare our results, we manually generated the ground truth RBNs. For each image, we chose between 1 and 6 RBNs that are fully visible (not occluded) and whose size exceeds the minimal dimensions mentioned above. The algorithm was implemented in Matlab, except for the face detection (we used an OpenCV implementation [7].)[1]

To quantitatively evaluate our results, we measured the *precision*, i.e., the percentage of correctly recognized RBNs out of the total number of recognized RBNs, and *recall*, i.e., the percentage of correctly recognized RBNs out of the total number of (ground-truth) RBNs in the set. We used the standard F score to combine the precision and recall into a single measure, that is,

$$F = \frac{Prec. \cdot Rec.}{\alpha \cdot Rec. + (1-\alpha) \cdot Prec.}. \quad (1)$$

The relative weight α was set to 0.5.

Note that several parameters are used in our system, such as minimum size of the face, maximum allowed stroke width, and allowed tag size relative to the face scale. We used fixed parameters in all the experiments.

3.1 Comparison to Conventional Approaches

In this experiment, we evaluated the contribution of each component in our pipeline. To do so, we compared our full pipeline (**“Face Detector+E-SWT+OCR”**) to the following two sub-systems:

- **“SWT+OCR”:** the SWT is applied on the entire image to locate the RBNs, followed by standard OCR on the detected text regions.

[1] Our datasets and the code are publicly available.

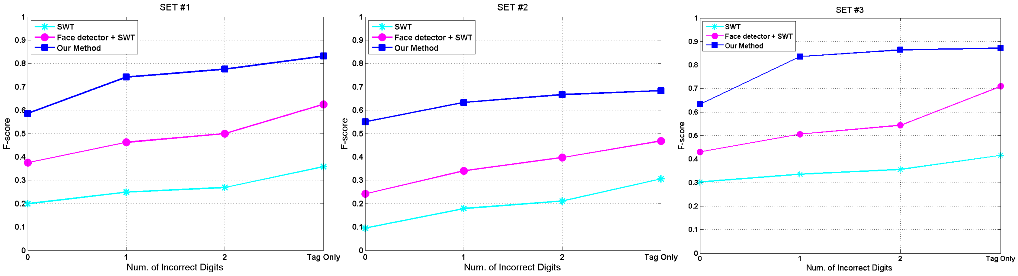

Figure 7: The first system (SWT+OCR) consists of SWT text detector, followed by OCR. In the second system (Face detector+E-SWT+OCR) the face detector is added, and the enhanced SWT (E-SWT) is used. Our method consists of the full pipeline (see Fig. 1). The graphs show the computed F-score for each dataset, for the cases of perfect detection, one wrong digit, two wrong digits, and only tag (the tag area is correctly detected but with more than 2 wrong digits).

- **"Face Detector+E-SWT+OCR":** the face detector is added to generate hypothesis search regions (see Sec.2.1). The SWT combined with our enhancements (noted by E-SWT) is applied on the hypothesis regions (see Sec. 2.2), followed by standard OCR.

To evaluate the performance, we computed the F-score for four levels of "correct-recognition": Perfect match (i.e., 0 incorrect digits), one wrong digit, two wrong digits, and only tag (more than two wrong digits). Fig. 7 presents the results of the two sub-systems compared to our full pipeline on each of the datasets. Comparison of the performance of the two sub-systems shows the contribution of limiting the search space according to the detected face as well as the use of E-SWT (improvement of $13\% - 29\%$). The average face detection recall rate on our dataset is 85%. Comparison between our full system and the second system isolates the contribution of the digit processing step (see Sec. 2.3) and the final step of filtering wrong detections (improvement of $16\% - 33\%$).

3.2 Comparison to End-to-End Scene Text Recognition (PLEX)

We compared our performance to the scene text recognition algorithm, ”PLEX" [12]. PLEX requires a fixed lexicon of words from which the text is recognized. Therefore, we generated two lexicons for each of our datasets: one lexicon consists of all the ground-truth RBNs in the relevant dataset (the lexicon sizes are given in Table 1), while the other consists of the ground truth plus 20% random RBNs. Note that these lexicons represent only a limited set of all possible RBNs. Therefore, the possible recognition errors are significantly reduced. Table 2 shows the precision, recall and F score of PLEX results (second and third rows) and

	Set #1			Set #2			Set #3		
	Prec.	Rec.	F	Prec.	Rec.	F	Prec.	Rec.	F
Our results	**0.66**	**0.50**	**0.57**	**0.75**	**0.45**	**0.56**	**0.65**	**0.62**	**0.63**
PLEX only GT [12]	0.41	0.40	0.41	0.67	0.45	0.54	0.51	0.32	0.39
PLEX GT+Dis. [12]	0.38	0.39	0.39	0.58	0.44	0.50	0.25	0.27	0.26
CARMEN [5]	0.67	0.37	0.48	0.68	0.38	0.49	0.73	0.47	0.57

Table 2: **RBNR Performance:** The computed precision, recall and F-score (w.r.t the ground truth) of our results compared with the results of PLEX [12] and CARMEN LPR system [5].

our method (first row). As can be seen, our method which did not make use of the lexicons, achieve better performance than PLEX. Restricting the set of possible RBNs is expected to improve our results.

3.3 Comparison to LPR

In this experiment, we compared our performance to the CARMEN FreeFlow LPR system [5]. CARMEN in a world leading commercial general-purpose LPR system, developed by A.R. Hungary. CARMEN FreeFlow algorithm provides high rate plate recognitions in a large variety of image scenes and plate types. The CARMEN's system is based on the visual characteristics of car license plates. To adapt it for our purpose, a special parameters adjustment was required. The performance of CARMEN on our datasets is shown in Table 2. For each dataset, the precision, recall and F-score are measured for perfectly correct recognition (i.e., all digits are correct). The results indicate that our system achieved higher F-score than CARMEN (we achieved higher recall than CARMEN and similar precision rate).

4 Conclusion

In this paper we presented a dedicated system for RBN recognition in natural image collections taken in running sport events. This specific problem falls between the problem of text detection under very restrictive conditions such as LPR, and the wider context of text detection in completely unstructured scenes (i.e., text detection in the "wild"). We showed that by using the knowledge that RBN is located on a person, enhancement of the SWT and proper processing of the detected tags, we achieved reliable and efficient recognition. The ideas demonstrated in the paper might be useful for other text detection problems in other partially-restricted scenarios. As far as limitations go, we clearly depend on the quality of the face detection. In addition, handling partial occlusions in the tag and better treatment of the non-rigidity of the tag are left for future research.

5 Acknowledgment

This work was supported in part by European Community grant PIRG05-GA-2009-248527.

References

[1] C.N.E. Anagnostopoulos, I.E. Anagnostopoulos, I.D. Psoroulas, V. Loumos, and E. Kayafas. License plate recognition from still images and video sequences: A survey. *Transactions on Intelligent Transportation Systems*, 2008.

[2] Xiangrong Chen and Alan L. Yuille. Detecting and reading text in natural scenes. In *CVPR*, pages 366–373, 2004.

[3] A. Coates, B. Carpenter, C. Case, S. Satheesh, B. Suresh, T. Wang, D.J. Wu, and A.Y. Ng. Text detection and character recognition in scene images with unsupervised feature learning. *ICDAR*, 2011.

[4] B. Epshtein, E. Ofek, and Y. Wexler. Detecting text in natural scenes with stroke width transform. *CVPR*, 2010.

[5] CARMEN FreeFlow. Number plate recognition products. http://www.arhungary.hu/040930/aloldalak/products/freeflow/frame.htm.

[6] Keechul Jung, Kwang In Kim, Anil K. Jain, and Anil K. Jain. Text information extraction in images and video: a survey. pages 977–997, 2004.

[7] R. Lienhart and J. Maydt. An extended set of haar-like features for rapid object detection. In *Image Processing. 2002. Proceedings. 2002 International Conference on*, volume 1, pages I–900. Ieee, 2002.

[8] L. Neumann and J. Matas. Text localization in real-world images using efficiently pruned exhaustive search. *ICDAR*, 2011.

[9] Lukas Neumann and Jiri Matas. A method for text localization and recognition in real-world images. In *ACCV*, pages 770–783, 2010.

[10] R. Smith. An overview of the tesseract ocr engine. *ICDAR*, 2007.

[11] K. Wang and S. Belongie. Word spotting in the wild. *ECCV*, 2010.

[12] K. Wang, B. Babenko, and S. Belongie. End-to-end scene text recognition. *ICCV*, 2011.