A Dataset for Arabic Text Detection, Tracking and Recognition in News Videos– AcTiV

Oussama Zayene[1, 2], Jean Hennebert[1, 3], Sameh Masmoudi Touj[2], Rolf Ingold[1] and Najoua Essoukri Ben Amara[2]

[1]DIVA group
Department of Informatics, University of Fribourg (Unifr)
Fribourg, Switzerland
[3]Institute of Complex Systems
HES-SO, University of Applied Science Western Switzerland
{oussama.zayene, jean.hennebert, rolf.ingold}@unifr.ch

[2]SAGE lab
National Engineering School of Sousse (Eniso), University of Sousse, Tunisia
najoua.benamara@eniso.rnu.tn
samehmasmouditouj@yahoo.fr

***Abstract*—Recently, promising results have been reported on video text detection and recognition. Most of the proposed methods are tested on private datasets with non-uniform evaluation metrics. We report here on the development of a publicly accessible annotated video dataset designed to assess the performance of different artificial Arabic text detection, tracking and recognition systems. The dataset includes 80 videos (more than 850,000 frames) collected from 4 different Arabic news channels. An attempt was made to ensure maximum diversities of the textual content in terms of size, position and background. This data is accompanied by detailed annotations for each textbox. We also present a region-based text detection approach in addition to a set of evaluation protocols on which the performance of different systems can be measured.**

Keywords—Video OCR; Video database; Benchmark; Arabic text

I. INTRODUCTION

The number of standard data sets has increased significantly over the last decade in all scientific research fields. This is explained by a variety of fundamental requirements, ranging from the development and evaluation of specific approaches to the need of systematic benchmark of systems using the same data set. This is further complemented by organization of international competitions to evaluate the performance of participants' algorithms under the same experimental conditions and protocols. In addition, the existence of such data sets saves researchers from the manual labeling of thousands of data which is a time-consuming task. Thus, the importance of having a publicly accessible annotated data set is widely recognized by the computer vision community.

Texts appearing in video sequences, especially captions, are one of the most important high-level information of the multimedia content. They carry vital search information for automatic video analysis and can be used as powerful semantic clues in video content retrieval. Hence, automatic extraction and reading of text from videos and natural scene images is a very active research area nowadays, such as in [4, 6-14]. As a result, the domain has developed a number of standard data sets covering different problem areas. Two main types of databases are considered. The first one focuses on text embedded in real scene images like SVT [8], MSRA-TD500 [9] and KAIST [10] databases. The second one focuses on text displayed in video sequences for which the first public video database was introduced in the ICDAR 2013 Robust Reading Competition (Challenge 3) [4]. In fact, the difference between these two sorts of databases is that the first one deals with a single input image, while the second deals with a set of consecutive frames that gives redundant temporal information. The large amount of temporal information can be used to improve the performances of text detection process [12]. It can also be exploited in several other tasks, such as video text tracking [6, 7, 13]. Although this information is potentially useful for text detection and tracking in consecutive video frames, there is still few benchmark databases for text in videos, probably due to the great effort that is needed especially for the ground truth annotation.

Our contribution concerns specifically the automatic extraction and recognition of texts from Arabic news video. Analysis of Arabic documents and recognition of Arabic texts became an attracting research domain in the recent years. Major contributions have already been made in the field of printed and handwritten OCR systems. Much of the progress is due to the availability of public data sets, such as the IFN/ENIT database [2] of Arabic handwritten words, ADAB database [5] of segmented online handwritten Arabic characters and the APTI database [1] which is a large-scale benchmark for printed text recognition. However, to our knowledge, no attempts have yet been made on the development of standard data sets for Arabic text in videos, despite the existence of more than 65 Arabic news channels around the world.

Considering this, we developed a new dataset of video sequences containing artificial Arabic text. The database has been named AcTiV for Arabic Text in Video. We have mainly targeted Arabic OCR systems and text detection systems which require the text transcription and the coordinates of all text regions as ground truth data. The challenges that are addressed by AcTiV are in text patterns variability (colors, fonts, sizes, position, etc.) and in the presence of complex background with various objects similar to text characters. AcTiV enables users to test their systems' abilities to locate, track and read text objects in videos. We also propose a set of evaluation protocols on which the performance of text detection, tracking and recognition algorithms can be compared and measured.

978-1-4799-1805-8/15/$31.00 ©2015 IEEE

In Section II of this paper, we describe the content of the dataset including details of the ground truth annotations. The evaluation protocols are described in section III. We then briefly present a region-based text detection approach in section IV and we report on its application to a subset of AcTiV in section V. Conclusions are given in section VI.

II. ACTIV-DATABASE DESCRIPTION

The AcTiV[1] dataset is developed, in cooperation between different labs to advance the research and development of video text detection and recognition systems.

A. Data Acquisition

TV channels broadcast a wide variety of programs across a range of genres including talk shows, interviews, documentaries, weather report and sports. News reports were specifically chosen for the present study. In order to ensure maximum diversities of the content and avoid repetition, recordings from the same channel were spaced by one week.

The broadcast streams were captured from a Direct Broadcast Satellite (DBS) system. The video stream was initially saved unaltered on the hard drive (MPEG-TS).Then, a transcoding process took place to convert the interlaced video to a de-interlaced MPEG4-AVC using an x264 based encoder and applying a YADIF filter. The goal of this process is both to prepare the video to frame-by- frame analysis and to lower the video bitrate without perceived quality loss. In the present work, two types of video stream were chosen: Standard-Definition (720x576, 25fps) and High-Definition (1920x1080, 25fps).

B. Characteristics and Statistics

The dataset includes 80 videos collected from 4 different Arabic news channels, including one HD channel. This choice is based on the fact that it ensures maximum diversities of the textual content in terms of font, size, position and background.

Fig. 1. Typical video frames from the proposed dataset. Sub-figures on the left: examples of Russia Today (RT) and ElWataniya 1 frames. These 2 channels do not contain scrolling text. Sub-figures on the right: examples of Aljazeera HD and France 24 frames.

[1] http://www.sage-eniso.org/content/fr/20/activ-data-base.html

We mainly focus on text displayed as overlay in news video, which can be classified into two types: static and dynamic (also called scrolling text). A text that does not undergo a change in its content, position, size, or color within its display interval is considered as static text. This group usually includes event information, speaker's name, subtitles, etc. Dynamic text targeted in our study, refers to the horizontal scrolling text that usually resides in the lower third of the television screen. In Arabic channels, dynamic text moves from left-to-right. Figure 1 shows some examples of static and scrolling texts from four Arabic news channels.

Each video is around 3 to11 minutes long. The maximum number of textboxes in one clip is 69. However, if we regard the same textbox across multiple frames as separate textboxes, we have an average of 6000 textboxes per clip. More statistical details are presented in Table I.

TABLE I. STATISTICS OF THE PROPOSED DATABASE

	Duration (mn)	No. of Frames	No. of textboxes	No. of textlines	No. of words
Aljazeera HD	152:28	228700	798	1029	4920
France 24 Arabic	101	150425	780	956	4875
Russia Today Arabic	121	181500	1387	1541	6040
El Wataniya 1	200:21	300525	1100	1298	5685
Overall	574:49	861150	4065	4824	21520

Additionally, the proposed dataset includes 5 different fonts with various sizes.

C. Ground truth Annotations

Generally, any standard database should undergo manual or semi-automatic annotation for ground truthing. In our case, we used AcTiV-GT Software [3] to semi-automatically annotate our collection of dataset.

Fig. 2. The user interface of AcTiV-GT software displaying a labeled frame

The annotation process consists of two different levels:

The global annotation, which concerns the entire video, is performed manually thanks to a user interface (shown in Figure 2). We first open a video clip, and then we draw a rectangle for each static text. Once a textbox has been selected, a new set of information is created. It contains the following elements:

- Time stamps for its apparition interval: start/end frame.

- Rectangle's attributes: (x, y) coordinates, width, height.
- Content data: text, text color, background color, background type<transparent, opaque>.

An extract of a global xml file is illustrated in Figure 3.

```
<?xml version="1.0" encoding="UTF-8" ?>
- <video id="3" channel="AljazeeraHD" resolution="1080p" duration="00:08:15"
  fps="25" nbOfFrames="12375">
  - <staticText nbOftextBox="49" font="aljazeeraFont">
    - <textBox id="1" nbOfaInterval="1">
        <aInterval id="1" frame_S="134" frame_E="376" />
        <position x="482" y="889" width="912" height="90" />
      - <content nbTextLines="1" textColor="252,252,250" bgColor=""
          bgType="transparent">
          <textLine id="1" transcription="ترحيب بالدستور التونسي الجديد"
            transcriptionLabel="Taaa_B Raa_E Haaa_B Yaa_M Baa_E Space
            Baa_B Alif_E Laam_B Daal_E Siin_B Taaa_M Waaw_E Raa_I
            Space Alif_I Laam_B Taaa_M Waaw_E Nuun_B Siin_M Yaa_E
            Space Alif_I Laam_B Jiim_M Daal_E Yaa_B Daal_E" />
        </content>
      </textBox>
    - <textBox id="2" nbOfaInterval="1">
        <aInterval id="1" frame_S="400" frame_E="625" />
        <position x="602" y="872" width="775" height="90" />
      - <content nbTextLines="1" textColor="252,252,250" bgColor=""
          bgType="transparent">
          <textLine id="1" transcription="كريستوفر رُسن يزور المغرب"
            transcriptionLabel="Kaaf_B Raa_E Yaa_B Siin_M Taaa_M Waaw_E
            Faa_B Raa_E Space Raa_I SiinChadda_E Space Yaa_B Zaay_E
            Waaw_I Raa_I Space Alif_I Laam_B Miim_M Ghayn_M Raa_E
            Baa_I" />
        </content>
      </textBox>
```

Fig. 3. Example of global XMLfile: part of the static text. This figure includes ground truth information about 2 textboxes from a total of 49.

Dynamic text is formed by continuous scrolling series of tickers. To annotate this kind of text, we noted for each ticker: its content, the first frame where the ticker appears and the initial offset in the first frame which is estimated using a virtual line. This information is stored in the global xml file as depicted in Figure 4.

```
- <scrollingText orientation="left-right" textColor="251,251,255"
    bgColor="@virginTicker(Image)" bgType="opaque" runningSpeed="6,770
    pixel/frame">
    <bandPosition x="0" y="977" width="1432" height="65" />
  - <content nbOftickerInformation="56">
      <tickerInformation id="1" frame_S="252" offset="4" transcription="مصر:
        المجلس الأعلى للقوات المسلحة يفوض السيسي الترشح للرئاسة"
        transcriptionLabel="Miim_B
        Saad_M Raa_E Colon Space Alif_I Laam_B Miim_M Jiim_M Laam_M
        Siin_E Space Alif_I Laam_EHamzaAboveAlif_E Ayn_B Laam_M
        AlifBroken_E Space Laam_B Laam_M Gaaf_M Waaw_E Alif_I Taaa_I
        Space Alif_I Laam_B Miim_M Siin_M Laam_M Haaa_M TaaaClosed_E
        Space Yaa_B Faa_M Waaw_E Daad_I Space Alif_I Laam_B Siin_M
        Yaa_M Siin_M Yaa_E Space Alif_I Laam_B Taaa_M Raa_E Shiin_B
        Haaa_E Space Laam_B Laam_M Raa_E HamzaAboveAlifBroken_B
        Alif_E Siin_B TaaaClosed_E" />
      <tickerInformation id="2" frame_S="443" offset="0" transcription="المجلس
        العسكري: لم يكن يوسفنا إلا الاستجابة لرغبة الجماهير في ترشيح السيسي"
        transcriptionLabel="Alif_I
        Laam_B Miim_M Jiim_M Laam_M Siin_E Space Alif_I Laam_B Ayn_M
        Siin_M Kaaf_M Raa_E Yaa_I Colon Space Laam_B Miim_E Space
        Yaa_B Kaaf_M Nuun_E Space Baa_B Waaw_E Siin_B Ayn_M Nuun_M
        Alif_E Space HamzaUnderAlif_I Laam_EAlif_E Space Alif_I
        Laam_EAlif_E Siin_B Taaa_M Jiim_M Alif_E Baa_B TaaaClosed_E
        Space Laam_B Raa_E Ghayn_B Baa_M TaaaClosed_E Space Alif_I
        Laam_B Jiim_M Miim_M Alif_E Haa_B Yaa_M Raa_E Space Faa_B
        Yaa_E Space Taaa_B Raa_E Shiin_B Yaa_M Haaa_E Space Alif_I
        Laam_B Siin_M Yaa_M Siin_M Yaa_E" />
```

Fig. 4. Example of global XMLfile: part of the dynamic text. This figure illustrates ground truth data about 2 tickers from a total of 56, and channel specific information (e.g. *runningSpeed* and *bandPosition*).

Arabic letters can be written in different shapes depending on their position in the word. In order to have an easily accessible representation of Arabic text for future processing, it is transformed into a set of labels with a suffix that refers to the letter's position in the word (B: Begin, M: Middle, E: End and I: Isolate). We use the same labels used in the APTI database [1] to standardize the character labels for Arabic text.

A transcription label is generated for each Arabic text stored in the xml file (static or dynamic); it is saved under the attribute *transcriptionLabel* within the same element that contains the Arabic text (shown in Figures 3 and 4).

In addition to these data, other information are stored in the global xml file such as the duration, the total number of textboxes in the video, the text font, the number of tickers, the band position etc.

The local annotation at the frame level is done automatically according to the information contained in the global metafile. For more details about the annotation framework please refer to our previous work [3].

III. PERFORMANCE EVALUATION

A. Metrics

Detection metrics: The performance of a text detection system is traditionally evaluated using the well-known precision and recall metrics that are defined as:

$$precision = \frac{\sum_{i=1}^{|D|} matchD(Di)}{|D|}$$

$$recall = \frac{\sum_{i=1}^{|G|} matchG(Gi)}{|G|}$$

Where D is the list of detected rectangles and G is the list of groundtruth rectangles. In the ICDAR'03 and ICDAR'05 competitions, the matching functions (*matchG* and *matchD*) only consider one-to-one matches between groundtruth and detected rectangles, resulting in ambiguity between detection quantity and quality. In [15] Wolf et al. redesigned the former matching functions considering different matching cases: one-to-one matching, one-to-many matching and many-to-one matching. The proposed performance measure was adopted in ICDAR'11 and ICDAR'13 competitions.

One-to-one match one-to-many match many-to-one match

Fig. 5. Different match types between ground truth rectangles (red lines) and detected rectangles (green lines).

Because bounding rectangles coordinates are the only information needed by these metrics, they can be easily applied to Arabic text.

Tracking metrics: In our work, the main task of text tracking scheme is to determine text apparition interval (text appearing/disappearing frames) for static text as well as scrolling one. In the ICDAR'13 competition [4] a set of metrics from the VACE framework [16] is used to measure the performance of the tracking systems. This set includes: the Multiple Object Tracking Precision *(MOTP)*, Multiple Object Tracking Accuracy *(MOTA)* and Average Tracking Accuracy *(ATA)*. These metrics provide spatio-temporal measures taking into account the number of correctly detected and tracked text objects, false positives and fragmentations.

Recognition metrics: The performance measure for the recognition task is based on insertion *(I)*, deletion *(D)* and

substitution *(S)* errors at the word level. The Word Error Rate *(WER)* is calculated as follows:

$$WER = \frac{I + S + D}{N}$$

Where N indicates the total reference words. On each word, the Character Error Rate *(CER)* is also computed. Figure 6 shows an example of *WER* metric scoring for a synthetic output.

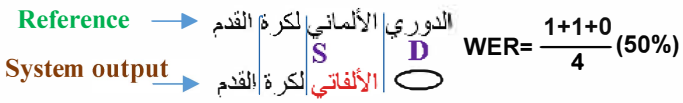

Fig. 6. Example of WER computation

B. AcTiV Protocols

A set of evaluation protocols is proposed (see Table II) taking advantage of the variability in data content. These different protocols enable researchers to test their algorithms under different situations.

TABLE II. ACTIV EVALUATION PROTOCOLS

Protocol	Resolution (Channel)	Type of Text Instances (Motion/Background)	Task
1	1920x1080 (AljazeeraHD)	static/complex static/simple	D
2		static/simple static/simple scrolling/simple	D/T
3		static/complex static/simple	R
4	720x576 (RT Arabic)	static/complex static/simple	D
5		static/complex static/simple	D/T
6		static/complex static/simple	R
7	All	static/complex static/simple	D
8		static/simple static/complex scrolling/simple	D/T
9		static/complex static/simple	R
10	All	static/simple static/complex scrolling/simple	End-to-End
11	All		TV Logo D

Protocol 1 aims to measure the performance of single-frame based methods to localize text regions in still HD images. A sub-dataset of non-redundant frames is created from the AcTiV-DB and used in this protocol. Detection ground truth is provided at the line level for each frame.

Protocol 2 focuses on static and scrolling text detection and tracking methods in HD videos. This protocol requires that text lines are both detected correctly in every frame and tracked correctly over the video sequence.

Protocol 3 aims to evaluate the performance of text recognition systems. A sub-dataset of cropped text images is created from the AcTiV-DB and used in this protocol.

Protocol 4, 5, 6 are similar to protocols 1, 2, 3 respectively, differing only by the channel resolution. All SD sequences in our database can be targeted by these protocols.

Protocol 7, 8, 9 are the generic version of the previous protocols where text detection, text tracking and text recognition tasks are evaluated independently to data quality.

Protocol 10 aims to evaluate the performance of end-to-end systems (simultaneous detection, tracking and recognition of all text lines in the video sequences).

Protocol 11 is meant for TV logo identification in video sequences. Although it is unrelated to previous protocols, it can be very helpful as a pre-processing step for other tasks to select the corresponding system depending to the channel.

IV. TEXT DETECTION APPROACH

As an application to the AcTiv-Database, we are currently developing an automated end-to-end text detection, tracking and recognition system. We focus in this work on text detection in video frames. The proposed approach mainly consists of four stages: (1) component extraction, (2) component filtering, (3) merging process and (4) text line formation and refinement.

Component extraction: To extract connected components (CCs) from an input image the Stroke Width Transform (SWT) method [11] is adopted for its effectiveness. SWT runs on edge map, so we firstly perform edge detection on the original video frame using the Canny edge detector. Then, the gradient direction d_p of each edge pixel p is determined. A pixel ray starting from p in the d_p direction is generated, and the first edge pixel q along the ray is located. If p and q have nearly opposite gradient directions, all the pixels inside the ray are labeled by the distance between p and q (called stroke width: SW). The next step in this stage is to group neighboring pixels in the resulting SWT image into CCs. In order to allow smoothly varying SWs in a letter, adjacent pixels are grouped if their SW ratio is less than 3.0. In Arabic script a single character may consist of several strokes and, subsequently, several labels. Considering this, we modified the original CC labelling operation [11] using a two-pass algorithm.

Component filtering: At this stage, we design a set of heuristic rules based on statistical and geometric proprieties of the components, to filter out CCs that are unlikely parts of texts. First of all we remove components with very large and very small aspect ratio under a conservative threshold so that characters like Alif "ا" are not discarded. Then we discard objects with unusual size by limiting the length and width of the component. Added to that, objects located at the border of the image will not be taken in account in further processes.

Merging process: Different from English text, an Arabic character may consist of several detached parts such as Hamza above/bellow Alif: "أ،إ", or Tild above Alif: "آ", or diacritic marks like dots. Among the previously obtained candidate CCs, some of them are parts of a character, which need to be merged into a single one bounding box. We design a small set of rules to group the CCs: (1) The CCs should have similar SW (ratio between the median SWs has to be less than 2.0). (2) Two near CCs with their centers in the same vertical line can be merged. (3) Two overlapping bounding boxes of suitable size should be merged.

Text line formation and refinement: In order to correctly form a text line out of a huge set of components, we define a probability matrix M, which for two different letter candidates C_i and C_j, $M_{i,j}$ is their corresponding matching probability. C_i and C_j are paired only if their merging probability is high enough (higher than a predefined threshold: T_m). For this reason we define four probability scores: (1) the closer C_i and C_j are, the more important $Ds(C_i, C_j)$ is. (2) Since text always appears in the form of straight lines, $Al(C_i, C_j)$ increases depending on components' alignment. (3) $Sw(C_i, C_j)$: probability based on SW similarity. (4) $Ov(C_i, C_j)$: probability based on spatial overlap between their corresponding rectangles. The probability matrix M is then calculated as follows:

$$M_{ij} = \begin{cases} 1 & if\ Ov(C_i, C_j) > T_{ov} \\ s & if\ Ds(C_i, C_j) > T_{ds}\ and \\ & \quad Al(C_i, C_j) > T_{al}\ and \\ & \quad Sw(C_i, C_j) > T_{sw} \\ 0 & otherwise \end{cases}$$

where

$$s = \frac{Ds(C_i\ C_j) + Al(C_i, C_j) + Sw\ (C_i\ C_j)}{3}$$

And T_{ov}, T_{ds}, T_{al} and T_{sw} are probability thresholds over the overlap ratio, distance, alignment and stroke width, respectively. Text lines formation process consists in pairing C_i and C_j where $M_{i,j}$= max(M) with respect to T_m threshold. The process ends when no components can be grouped.

During the previous merging process false positives can be grouped resulting a large number of false text lines. Therefore as a refinement step, we use the well-known projection-profile, text contrast and aspect ratio, since text appears in horizontal direction and has high contrast compared to its background.

V. EXPERIMENTS

In order to evaluate the effectiveness of the proposed text detection approach, we developed our own evaluation tool based on [15] with few enhancements to support the specificities of our dataset in terms of groundtruth information and data format. Under **protocol 1**, a total of 425 frames are used as benchmark. The algorithm was able to detect captions on simple and complex backgrounds, text with various colors and sizes, and low contrast text. In our experiments we used the area precision/recall thresholds proposed in the original publication [15]: t_p = 0.4 and t_r = 0.8. All the classes in the proposed system were coded and compiled using Java 1.8.

TABLE III. EVALUATION OF TWO TEXT DETECTION METHODS

Data	Method	Recall	Precision	F-measure
425 frames	Our Method	0.67	0.73	0.70
	Epshtein [11]	0. 50	0. 30	0. 40

As shown in table III, our method outperforms the SWT-based method [11] that already shown good performance compared to several other existing methods.

VI. CONCLUSION

In this paper, we presented the new AcTiV dataset for the development and evaluation of text detection, tracking and recognition systems targeting Arabic news video. This dataset is freely available to research institutions. We provided details about the data acquisition, the characteristics and statistics of the database. We also reported about our ground truthing software used to semi-automatically annotate the video clips.

We evaluated a text detection algorithm as proof-of-concept of the new dataset. Additionally, a set of evaluation was made to measure systems' performance under different situations. The database will probably evolved with new TV channels and additional ground truth information such as "don't care" regions.

REFERENCES

[1] F. Slimane, R. Ingold, S. Kanoun, A. M. Alimi and J. Hennebert, "A New Arabic Printed Text Image Database and Evaluation Protocols", International Conference on Document Analysis and Recognition (ICDAR), July 2009.

[2] M. Pechwitz, S. Maddouri, V. Maergner, N. Ellouze, H. Amiri, "IFN/ENIT database of handwritten Arabic words", Colloque International Francophone sur l'Écrit et le Document (CIFED), October 2002.

[3] O. Zayene, S. M. Touj, J. Hennebert, R. Ingold and N. E. Ben Amara "Semi-Automatic News Video Annotation Framework for Arabic Text", Image Processing Theory, Tools and Applications (IPTA), October 2014.

[4] D. Karatzas et al., "ICDAR 2013 Robust Reading Competition", (ICDAR), August 2013.

[5] H. El Abed, V. Margner, M. Kherallah and A. M. Alimi "ICDAR 2009 Online Arabic Handwriting Recognition Competition", (ICDAR), July 2009.

[6] L. Gomez and D. Karatzas, "MSER-based Real-Time Text Detection and Tracking", International Conference on Pattern Recognition (ICPR), August 2014.

[7] W. Huang, P. Shivakumara and C. L. Tan, "Detecting moving text in video using temporal information", (ICPR), December 2008.

[8] K. Wang and S. Belongie, "Word Spotting in the Wild", The 11th European Conference on Computer Vision (ECCV), September 2010.

[9] C. Yao, X. Bai, W. Liu, Y. Ma, Z. Tu, "Detecting texts of arbitrary orientations in natural images", In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2012.

[10] S. Lee, M. S. Cho, K. Jung, and J. Hyung Kim, "Scene Text Extraction with Edge Constraint and Text Collinearity", (ICPR), August 2010, Istanbul, Turkey (available at .http://ai.kaist.ac.kr/home/DB/SceneText).

[11] B. Epshtein, E. Ofek, and Y.Wexler, "Detecting text in natural scenes with stroke width transform", In Proc. (CVPR), June 2010.

[12] T. Lu Sh. Palaiahnakote C. Lim T. W. Liu, "Video Text Detection", Advances in Computer Vision and Pattern Recognition (ACVPR), july 2014.

[13] T. Yusufu,Y. Wang, X. Fang, "A Video Text Detection and Tracking System", IEEE International Symposium on Multimedia (ISM), December 2013.

[14] Q. Ye and D. Doermann, "Text Detection and Recognition in Imagery: A Survey", IEEE Transactions Pattern Analysis and Machine Intelligence (PAMI), November 2014.

[15] C. Wolf and J. M. Jolion, "Object count/area graphs for the evaluation of object detection and segmentation algorithms", International Journal on Document Analysis and Recognition (IJDAR), April 2006.

[16] R. Kasturi et al., "Framework for performance evaluation of face, text, and vehicle detection and tracking in video: Data, metrics, and protocol", IEEE Trans. (PAMI), February 2009.