2nd International Conference on Advanced Technologies
for Signal and Image Processing - ATSIP'2016
March 21-24, 2016, Monastir, Tunisia

Data, Protocol and Algorithms for Performance Evaluation of text detection in Arabic news Video

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

[1] SAGE group, National Engineering School of Sousse, Sousse, Tunisia
najoua.benamara@eniso.rnu.tn
samehmasmouditouj@yahoo.fr
[2] DIVA group, Department of Informatics, University of Fribourg, Fribourg, Switzerland
{firstname.lastname}@unifr.ch
[3] Institute of Complex Systems HES-SO, University of Applied Science Western Switzerland
jean.hennebert@hefr.ch

***Abstract*—Benchmark datasets and their corresponding evaluation protocols are commonly used by the computer vision community, in a variety of application domains, to assess the performance of existing systems. Even though text detection and recognition in video has seen much progress in recent years, relatively little work has been done to propose standardized annotations and evaluation protocols especially for Arabic Video-OCR systems. In this paper, we present a framework for evaluating text detection in videos. Additionally, dataset, ground-truth annotations and evaluation protocols, are provided for Arabic text detection. Moreover, two published text detection algorithms are tested on a part of the AcTiV database and evaluated using a set of the proposed evaluation protocols.**

Keywords—text detection; Evaluation Protocol; AcTiV database, Arabic Video-OCR

I. INTRODUCTION

Text appearing in videos often carries significant information such as name, place, events, etc. These semantic clues can be used in video content retrieval and indexing. To extract texts from video content, which is often called Video-OCR, the first essential step is to detect the text area in the video clip/frame. Comprehensive surveys of text detection in images and video can be found in [3, 17]. Evaluation algorithms are essential tools for researchers to compare their results with those of the literature. In order to evaluate a text detector three inputs are needed: video dataset, groundtruth (G) and system output (D), as shown in figure 3.

The state-of-the-art contain several available standard datasets with well-defined evaluation protocols for Latin and Chinese video texts detection as well as for real scene images. Examples include SVT [11] and KAIST [12] databases for scene text detection and ICDAR databases [2, 18] for text detection in videos. A good overview of the detection systems and their performance evaluation results can be found in the ICDAR Robust Reading competition [18]. However, Arabic video text detection remains a relatively unexplored field. The existing methods published in [14-16] are tested on private datasets with non-uniform evaluation protocols. In other words, direct comparison is not possible as these methods use different data and different evaluation protocols.

During the evaluation process G rectangles are compared to D ones based on a matching strategy. Final performance values are then calculated based on the well-known pixel-based or box-based precision and recall metrics. Previous related works such as [5, 6, 13] deal directly with evaluation of text detectors. In [13] Anthimopoulos et *al.* proposed an evaluation algorithm based on estimated number of characters in a bounding box. This number is approximated by the aspect ratio of the box based on the assumption that this ratio is invariable for every character and the spaces between words in a textline are proportional to its height. The precision and recall are then calculated based on area coverage and normalized by the approximation of the characters number for every rectangles. This method cannot be replicated for Arabic texts due to the presence of non-uniform intra/inter word distances and diacritic marks. In [6] Kasturi et *al.* proposed an overall metric for single frame-based text detection called Frame Detection Accuracy (FDA). This kind of evaluation methods is generally based on one-to-one matching between groundtruth and detected objects (the number of elements of both sets should be equal). However in text detection context, it is likely that the number of G rectangles may not be equal to the one of D rectangles (due to split/merge cases). Wolf and Jolion [5] used the main idea of Liang et *al.* [4] which was implemented for the evaluation of page document segmentation and adjusted it to the evaluation of text detection in videos. Liang et *al.* proposed the creation of overlap score matrices between every possible pair of blocks in order to assess the page segmentation algorithms. The advantage of this kind of algorithms is their capability to take into account the possible merge and split cases in addition to the one-to-one correspondence.

Motivated by the study in [4] and [5], we propose in this work a new fast and simple evaluation tool for artificial horizontally text detection in Arabic news video, in addition to an annotated database and a set of evaluation protocol. Two published text detectors are evaluated using our proposed tool.

This paper is organized as follows: detection dataset and evaluation protocol are presented in section II. In section III, we will describe the proposed evaluation algorithm. The text detection approach is described in section IV. Results are presented in Section V. Conclusion is given in section VI.

978-1-4673-8526-8/16/$31.00 ©2016 IEEE

II. DATASET & EVALUATION PROTCOL

AcTiV-DB is the first publicly accessible annotated dataset designed to assess the performance of different Arabic VIDEO-OCR systems [8]. Here bellow we briefly list some characteristics of the dataset:

- News reports were specifically chosen for the present database.
- AcTiV-DB includes 80 videos collected from four Arabic news channels: AljazeeraHD, France 24 arabic, Russia Today arabic and Elwataniya 1 TV (figure 1).
- The collected Videos are captured from a DBS system and converted to a de-interlaced MPEG4-AVC.
- Two different resolutions: Standard-Definition (720x576) and High-Definition (1920x1080).
- Texts are in five different fonts with various sizes.

The challenges that are addressed by AcTiV-DB are in text patterns variability (colors, fonts, sizes, position, etc.) and presence of complex background with various text-like objects. AcTiV-DB enables users to test their systems' abilities to locate, track and recognize text objects in videos.

Fig. 1. Frame samples extracted from video clips of the AcTiV dataset

A. AcTiV-D dataset

AcTiV-D (D for Detection) represents a sub-dataset of non-redundant frames collected from the AcTiV-DB and used to measure the performance of single-frame based methods to detect/localize text regions in still HD/SD images. AcTiV-D consists of 1843 frames (5133 textlines) distributed on four sets (one set per channel). Every set includes two sub-sets: trainingFiles and testFiles. More details are in table I.

TABLE I. STATISTICS OF THE ACTIV-*D* DATASET

Resolution	TV Channel	Training *textlines*	Test *textlines*
HD (1920x1080)	AljazeeraHD	803	226
SD (720x576)	France 24	960	224
	Russia Today	1302	317
	ElWataniya 1	1068	233

Detection groundtruth is provided at the line level for each frame. The XML format is an extended version of the format developed for the ICDAR Robust Reading Competition [2, 18]. Figure 2 shows an example for a part of a groundtruth xml file.

```
<?xml version="1.0" encoding="UTF-8"?>
- <Protocol4 channel="TunisiaNat1">
  - <frame id="7" source="vd01">
      <rectangle id="1" x="506" y="464" width="61" height="14"/>
      <rectangle id="2" x="66" y="499" width="491" height="32"/>
    </frame>
  - <frame id="16" source="vd01">
      <rectangle id="1" x="441" y="464" width="127" height="18"/>
      <rectangle id="2" x="373" y="499" width="184" height="27"/>
    </frame>
  - <frame id="64" source="vd01">
      <rectangle id="1" x="429" y="462" width="138" height="24"/>
    </frame>
```

Fig. 2. Example of groundtruth file for ElWataniya1 Channel (TunisiaNat1)

One bounding box is described by the element <Rectangle> which contains the rectangle's attributes: (x, y) coordinates, width and height. This xml file was generated by our Semi-automatic annotation framework published in [1]. The XML file is the same for the groundtruth and for the detection outputs. The output image and the groundtruth one must have the same label:
[channel_source_frame_id] (i.e. TunisiaNat1_vd01_frame_7).

To test systems' abilities to detect and locate texts under different situations, the proposed sub-dataset includes some frames which do not contain any text and some others which contain the same text regions but with different background.

B. AcTiV-D Protocols

A set of evaluation protocols is proposed by Zayene et *al.* [8] taking advantage of the variability in data content. In this paper, we focus only on the protocols dedicated to the detection task (task "D", in table II), the other protocols are not the subject of this paper.

TABLE II. ACTIV-*D* EVALUATION PROTOCOLS

Protocol	TV Channel		Type of Text Instances (Motion/Background)	Task
ptac_1	AljazeeraHD		static/complex static/simple	D
ptac_4	4.1	France 24	static/complex static/simple	D
	4.2	Russia Today		
	4.3	Elwataniya 1		
ptac_7	All		static/complex static/simple	D

Protocol 1 (ptac_1) aims to measure the performance of single-frame based methods to localize text regions in still HD images. *Protocol 4* (ptac_4) is similar to protocols 1, differing only by the channel resolution. All SD channels in our database can be targeted by this protocol. *Protocol 7* (ptac_7) is the generic version of the previous protocols where text detection is evaluated regardless of data quality.

III. EVALUATION ALGORITHM

The evaluation of text detection algorithms is generally based on two sets of information: a list G of ground truth text

rectangles and a list D of detected text rectangles. From these two lists, our goal is to generate a performance value for each text detector, as shown in figure 3. However, it's not simple as it seems. The generated performance value must depend on the following parameters:

- Quantity: number of correctly detected text objects, number of undetected text objects (miss detections) and number of false alarms.
- Quality: matching quality and splits or merges cases.

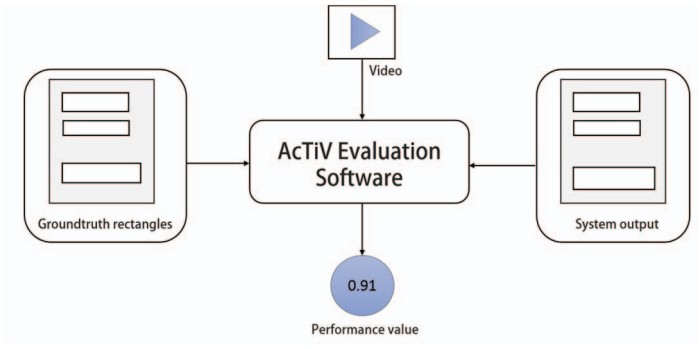

Fig. 3. AcTiV Evaluation software input

The most commonly used evaluation criteria are precision and recall. They are calculated by measuring the overlap between the intersection area of two rectangles (G_i , D_i) and the area of G_i (for recall) or D_i (for precision).

$$R_{AR}(G_i, D_j) = Area(G_i \cap D_j) \div Area(G_i) \quad (1)$$
$$P_{AR}(G_i, D_j) = Area(G_i \cap D_j) \div Area(D_j)$$

In order to match two sets of rectangles there are many optimized algorithms which had been developed in the past years [6, 20], but they take into account only the one-to-one matching cases. Therefore, our matching strategy is based on Liang et *al.* [4] and extended by Wolf and Jolion [5] which is fairly simple: from the two sets G and D we create two overlap matrices σ and τ as follows:

$$\sigma_{ij} = R_{AR}(G_i, D_j) \quad (2)$$
$$\tau_{ij} = P_{AR}(G_i, D_j)$$

To start matching we need to define two quality constraints, *tp* and *tr*, where $tp \in [0, 1]$ is the constraint on area precision and $tr \in [0, 1]$ is the constraint on area recall. Then we define the following constraints:

$$\sigma_{ij} > tr \quad \text{(a)} \quad (3)$$
$$\tau_{ij} > tp \quad \text{(b)}$$

So what we basically have now are two matrices which describe the recall and precision of |G| ground truth rectangles and |D| detected ones.

- If a rectangle G_i matches with a detected rectangle D_j this means that $\sigma_{ij} > tr$ and $\tau_{ij} > tp$ but that's not enough to conclude, these two constraints needs to be satisfied by only row *i* of the first matrix and column *j* of the second one. This is the case of **one-to-one matches** (Figure 4).
- If a rectangle G_i satisfies the constraint (3.a) with a detected rectangle D_j but does not satisfy the second one (3.b), (the recall respects the constraint but the precision does not), it means that the D_j could have merged a set S_n of groundtruth rectangles. This is the case of **many-to-one matches** (Figure 4).

This can be true only if the following conditions are respected:

$$\forall i \in S_n: \sigma_{ij} \geq tr \quad (4)$$
$$\sum_{i \in Sn} \tau_{ij} \geq tp$$

If it does not satisfy these two constraints, then Gi will be considered as undetected.

- If a rectangle D_j satisfies the constraint (3.b) with a ground truth rectangle G_i but does not satisfy the first one (3.a), (the precision respects the constraint but the recall does not), it means that the G_i could have been split into a set S_o of detected rectangles. This is the case of **one-to-many matches** (Figure 4).

This can be true only if the following conditions are respected:

$$\forall i \in S_o: \tau_{ij} \geq tp \quad (5)$$
$$\sum_{j \in So} \sigma_{ij} \geq tr$$

If it does not satisfy these two constraints, then D_i will be considered as a false alarm.

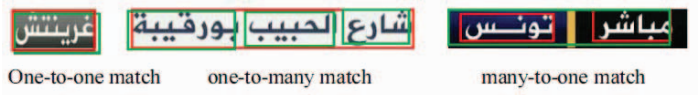

Fig. 4. Different match types between ground truth rectangles (red lines) and detected rectangles (green lines).

For this matching strategy, new measures have been defined:

$$R_{OB}(G, D, tr, tp) = \Sigma MatchG(G_i, D, tr, tp) \div |G| \quad (6)$$
$$P_{OB}(G, D, tr, tp) = \Sigma MatchD(D_j, G, tr, tp) \div |D|$$

Where $Match_G$ and $Match_D$ are functions that calculate the matching value depending on the quality of the match:

(7)

$$MatchG(Gi, D, tr, tp) =$$
$$\begin{cases} 1 & \text{if Gi matches against a single detected rectangle} \\ 0 & \text{if Gi does not matches against any detected rectangles} \\ f(k) & \text{if Gi matches against k detected rectangles} \end{cases}$$

(8)

$$MatchD(Dj, G, tr, tp) =$$
$$\begin{cases} 1 & \text{if Dj matches against a single ground truth rectangle} \\ 0 & \text{if Dj does not matches against any ground truth rectangles} \\ f(k) & \text{if Dj matches against k ground truth rectangles} \end{cases}$$

And f(k) is a parameter function which controls, during the evaluation process, the amount of punishment in case of scattering. In our case $f(k) = 1 \div (1 + ln(k))$ which corresponds to the index of fragmentation proposed by Mariano et *al.* [19].
In case of multiple images (video), we compare a set of lists *G* and *D* list by list, each list represents the entire rectangles G_k and D_k of a particular image. The new recall and precision are defined as the following:

(9)

$$R_{OB}(G,D,tr,tp) = \sum_k \sum_j Match_G\,(G_i^k, D^k, tr, tp) \div \sum_k |G^k|$$

$$P_{OB}(G,D,tr,tp) = \sum_k \sum_j Match_D\,(D_i^k, G^k, tr, tp) \div \sum_k |D^k|$$

However in order to evaluate an algorithm, a single performance value is required, i.e. the harmonic mean of the precision and recall measures (10).

$$Perf_{OB} = 2\,(R_{OB} \cdot P_{OB}) \div (R_{OB} + P_{OB}) \qquad (10)$$

Fig. 5. AcTiV Evaluation Software user interface.

Figure 5 shows the user interface of our evaluation software. The red text is the ground truth object rectangle, while the green one represents the detection result. The user can then apply the evaluation procedure to the current frame (by clicking on the "Evaluate CF" button) or all video frames (by clicking on the "Evaluate All" button). The "Performance Value" button displays precision, recall and F-measure values (see figure 6).

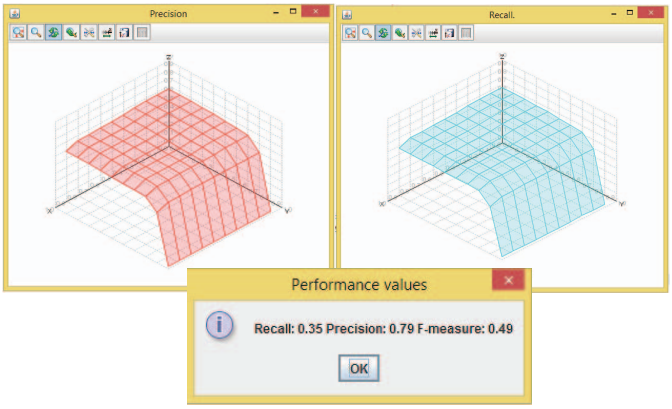

Fig. 6. AcTiV evaluation software output.

The precision and recall curves are depicted in Figure 6, where x-axis denotes *tr* values and y-axis denotes *tp* values (precision and recall values by varying *tr* and *tp* from 0 to1 by step of 0.1). This is helpful to choose good threshold values for your algorithm to say whether a rectangle had been correctly detected or not.

IV. Text Detection Algorithm

Text embedded in video frames often carries significant information such as place, name, events, etc. These semantic cues can be used in video content retrieval. To extract texts from video, which is often referred to as Video OCR, the first essential step is to detect/localize the text region in the video clip/frame. There are several published efforts addressing the problem of text area detection in images/video [3, 17].
The baseline algorithm that we present here is the work of Zayene et *al.* [8]. As shown in figure 7 the approach consists of 6 steps.

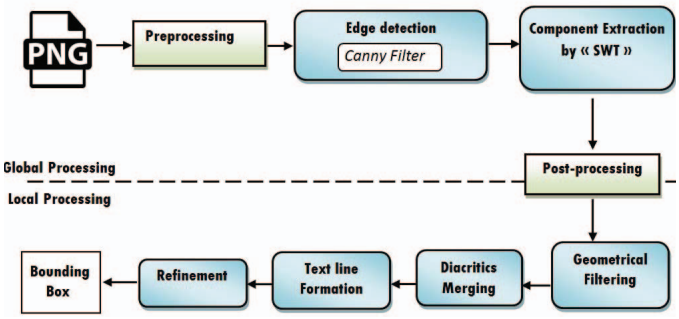

Fig. 7. Pipeline of the text detection algorithm. Two passes are performed, one for each text polarity (Dark text on Light background or Light text on Dark background)

A. Preprocessing and edge detection

The original frame is firstly converted to grayscale. Then, to compute the Stroke Width Transform (SWT) [7], an edge map and the X & Y gradients are required. Before calculating these, we blur the grayscale image to increase robustness against noise. For the edge map, we use a 3x 3 filter matrix and perform canny edge detection [9] with empirical thresholds of 175 and 320. For the X and Y gradients, we use the Sobel operator. The edge map is shown in figure 9-2.

B. Component Extraction by SWT

The SWT algorithm [7] is used to extract the connected components (CCs) from an input frame. This operator detects stroke pixels by shooting a search ray from an edge pixel p to its opposite edge pixel q along the gradient direction dg. If these two edge pixels have nearly opposite gradient orientations, the ray is considered valid. All pixels inside this ray are labeled by the distance between p and q.

In order to reduce the noises of incorrect connections produced by the SWT, we propose to discard the false rays whose length are higher than a predefined empirical threshold i.e., T_r. The neighboring pixels in the resulting SWT image are then grouped into CCs. In Arabic script a single character may consist of several strokes and, subsequently, several labels. Considering this, we modified the original CC-labelling operation [7] using a two-pass algorithm.

C. Component Analysis

Geometrical filtering: At this stage, we design a set of heuristic rules based on statistical and geometric properties 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. In addition, objects located at the border of the image will not be taken in account in further processes.

Diacritics merging: Different from English script, an Arabic character may consist of several diacritic marks such as Hamza above/bellow Alif: "أ،إ", or Tild above Alif: "آ", or dots. Among the previously obtained CC candidates, some of them are parts of a character, which need to be merged into a single bounding box. We design a small set of rules to group the CCs: (1) the CCs should have similar SW (ratio between the median SW values has to be less than 2.0). (2) The vertical distance between two CCs should not exceed an empirical predefined threshold i.e., T_{vd}.

D. Textline formation

In order to form the larger context of textual information, giving the obtained character/subword candidates, we develop a textline grouping method. Specifically, we define an upper triangular probability matrix M, where $m_{i,j}$ is the matching probability corresponding to a pair of text candidates (C_i, C_j). In order to compute $m_{i,j}$ for a given pair of components, we firstly calculate the following probability scores: *Ov(C_i, C_j)*: probability based on spatial overlap between their corresponding rectangles i.e., R_i, R_j, respectively. *Ds(C_i, C_j)*: probability based on the proximity of R_i and R_j. the closer R_i and R_j are, the more important Ds(C_i, C_j) is. *Al(C_i, C_j)* increases depending on components' alignment, since text always appears in the form of straight lines. *Sw(C_i, C_j)*: probability based on SW similarity.
The probability matrix *M* is then calculated as follows:

$$m_{ij} = \begin{cases} 1 & \text{if } \mathrm{Ov}(C_i, C_j) > T_{ov} \\ s & \text{if } \mathrm{Ds}(C_i, C_j) > T_{ds} \text{ and } \mathrm{Al}(C_i, C_j) > T_{al} \text{ and } \mathrm{Sw}(C_i, C_j) > T_{sw} \\ 0 & \text{otherwise} \end{cases} \quad (12)$$

Where

$$s = \frac{\mathrm{Ds}(C_i\, C_j) + \mathrm{Al}(C_i, C_j) + \mathrm{Sw}(C_i\, C_j)}{3} \quad (13)$$

And T_{ov}, T_{ds}, T_{al} and T_{sw} are probability thresholds over the overlap ratio, distance, alignment and stroke width scores, respectively. Text lines formation process consists finally in pairing C_i and C_j when $m_{i,j}$= max(M) with respect to a minimum matching probability threshold i.e, T_m. The process ends when no components can be grouped (see figure 8-g).

E. Refinement

In the previous text formation 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 (see figure 8-h).

V. Experimental Results

A. Parameter settings

In all these tests, the parameters of the proposed system were set empirically as follows. In the components extraction module: the maximum ray length value $\mathbf{T_r}$ = 60 px. In the geometrical filtering module: maximum character/subword height $\mathbf{h_{max}}$ = 40 px, character/subword width limit $\mathbf{w_{max}}$= 120 px and max aspect ratio $\mathbf{r_{max}}$ = 5. In the vertical merging module: maximum relative vertical distance $\mathbf{T_{vd}}$ = 3 px. Note that these values concern SD channels. In case of HD channels, they should be doubled. The probability thresholds, in the textline formation procedure, were set at these values: $\mathbf{T_{ov}}$=0.75, $\mathbf{T_{ds}}$=0.35, $\mathbf{T_{al}}$=0.35, $\mathbf{T_{sw}}$=0.24 and $\mathbf{T_m}$=0.5.

B. Results

To evaluate the proposed method, we compared it with another published text detector using our evaluation tool (described in section III). The results are given in Table III in terms of precision, recall and F-Measure.

TABLE III. Evaluation Results for procotols 1 and 4

Protocol	Method	Recall	Precision	F-measure
1	Our Method	0.69	0.73	0.71
	Epshtein [7]	0. 50	0. 30	0. 40
4.1	Epshtein [7]	0.51	0.32	0.41
	Our Method	0.62	0.7	0.66
4.2	Epshtein [7]	0.42	0.36	0.39
	Our Method	0.55	0.66	0.6
4.3	Epshtein [7]	0.47	0.35	0.41
	Our Method	0.71	0.68	0.69

In our experiments we used the area precision/recall thresholds proposed in the publication [5]: *tp* = 0.4 and *tr* = 0.8. As shown in table III, our method outperforms the method proposed in [7]. The algorithm was able to detect captions on simple and complex backgrounds, text with various colors and sizes, and low contrast text. All the classes in the proposed systems (the evaluation tool and the text detection approach) were coded and compiled using Java 1.8.

VI. Conclusions and Feature Work

In this paper, we presented three main ingredients for the Video-OCR domain. First, we proposed a new text detector evaluation system. Then, we gave a detailed description of the annotated sub-dataset AcTiV-D in addition to a set of evaluation protocols dedicated to video text detection systems. We also tested the performance of two published text detectors

Fig. 8. Text detection process. (a) Grayscale image, (b) Canny edges, (c) SWT dark-on-light text (we show only the first pass in this example), (d) CCs before filtering, (e) CCs after geometrical filtering, (f) Diacritics merging, (g) Textline formation and (h) Refinement (final resut)

using our evaluation tool. As a future work, we aim to extend our tool to an evaluation framework covering text detector, text tracker and OCR system evaluation.

REFERENCES

[1] 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.

[2] D. Karatzas et al., "ICDAR 2013 Robust Reading Competition", In Proc. Of the International Conference on Document Analysis and Recognition (ICDAR), August 2013.

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

[4] J. Liang, I.T. Phillips, and R.M. Haralick. Performance evaluation of document layout analysis algorithms on the UW data set. In Document Recognition IV, Proceedings of the SPIE, pages 149–160, 1997.

[5] 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.

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

[7] B. Epshtein, E. Ofek, and Y.Wexler, "Detecting text in natural scenes with stroke width transform", In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2010.

[8] O. Zayene, J. Hennebert, S. M. Touj, R. Ingold, and N. E. BenAmara, "A dataset for arabic text detection, tracking and recognition in news videos- AcTiV", ICDAR, August 2015.

[9] L. Ding, and A. Goshtasby. "On the Canny edge detector". Pattern Recognition (PR), 34(3), 721-725, 2001.

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

[11] C. Yao, X. Bai, W. Liu, Y. Ma, Z. Tu, "Detecting texts of arbitrary orientations in natural images", CVPR, June 2012.

[12] S. Lee, M. S. Cho, K. Jung, and J. Hyung Kim, "Scene Text Extraction with Edge Constraint and Text Collinearity", In Proceedings of the IEEE International Conference on Pattern Recognition (ICPR), August 2010, Istanbul, Turkey (available at .http://ai.kaist.ac.kr/home/DB/SceneText).

[13] M. Anthimopoulos, B. Gatos and I. Pratikakis "A two-stage scheme for text detection in video images", Image and Vision Computing, vol.28, 2010, pp.1413-1426.

[14] M. Ben Halima, A.M. Alimi, H. Karray and A. Fernandez Vila, "Nf-savo: Neuro-fuzzy system for arabic video ocr", Int. Journal of Advanced Computer Science and Applications, pp. 128–136, November 2012.

[15] [8] S. Yousfi, S.Berrani, C. Garcia, "Arabic text detection in videos using neural and boosting-based approaches: Application to video indexing", In Proceedings of the IEEE International Conference on Image Processing (ICIP), October 2014.

[16] [9] A. Jamil, I.Siddiqi, F. Arif and A. Raza, "Edge-based Features for Localization of Artificial Urdu Text in Video Images", ICDAR, September 2011.

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

[18] D. Karatzas et al., "ICDAR 2015 Robust Reading Competition", ICDAR, August 2015.

[19] V.Y. Mariano, J. Min, J.-H. Park, R. Kasturi, D. Mihalcik, H. Li, D. Doermann, and T. Drayer, "Performance Evaluation of Object Detection Algorithms", ICPR, 2002.

[20] S. M. Lucas, "ICDAR 2005 text locating competition results," in Proc. of ICDAR, September 2005.