Chapter 18
Benchmarking Strategy for Arabic Screen-Rendered Word Recognition

Fouad Slimane, Slim Kanoun, Jean Hennebert, Rolf Ingold, and Adel M. Alimi

Abstract This chapter presents a new benchmarking strategy for Arabic screen-based word recognition. Firstly, we report on the creation of the new APTI (Arabic Printed Text Image) database. This database is a large-scale benchmarking of open-vocabulary, multi-font, multi-size and multi-style word recognition systems in Arabic. Such systems take as input a text image and compute as output a character string corresponding to the text included in the image. The challenges that are addressed by the database are in the variability of the sizes, fonts and styles used to generate the images. A focus is also given on low resolution images where anti-aliasing is generating noise on the characters being recognized. The database contains 45,313,600 single word images totalling more than 250 million characters. Ground truth annotation is provided for each image from an XML file. The annotation includes the number of characters, the number of pieces of Arabic words (PAWs), the sequence of characters, the size, the style, the font used to generate each image, etc. Secondly, we describe the Arabic Recognition Competition: Multi-Font Multi-Size Digitally Represented Text held in the context of the 11th International Conference on Document Analysis and Recognition (ICDAR'2011), during September 18–21, 2011, Beijing, China. This first

F. Slimane (✉) · J. Hennebert · R. Ingold
DIVA Group, Department of Informatics, Universtity of Fribourg, Bd. de Perolles 90, 1700 Fribourg, Switzerland
e-mail: fouad.slimane@unifr.ch

J. Hennebert
e-mail: jean.hennebert@unifr.ch

R. Ingold
e-mail: rolf.ingold@unifr.ch

S. Kanoun
National School of Engineers (ENIS), University of Sfax, BP 1173, Sfax 3038, Tunisia
e-mail: slim.kanoun@ieee.org

A.M. Alimi
REGIM Group, National School of Engineers (ENIS), University of Sfax, BP 1173, Sfax 3038, Tunisia
e-mail: Adel.Alimi@ieee.org

V. Märgner, H. El Abed (eds.), *Guide to OCR for Arabic Scripts*,
DOI 10.1007/978-1-4471-4072-6_18,

edition of the competition used the freely available APTI database. Two groups with three systems participated in the competition. The systems were compared using the recognition rates at the character and word levels. The systems were tested on one test dataset which is unknown to all participants (set 6 of APTI database). The systems were compared on the ground of the most important characteristic of classification systems: the recognition rate. A short description of the participating groups, their systems, the experimental setup and the observed results are presented. Thirdly, we present our DIVA-REGIM system (out of competition at ICDAR'2011) with all results of the Arabic recognition competition protocols.

18.1 Introduction

It is universally acknowledged that more than 500 million people around the world speak and use Arabic as their liturgical language. Arabic is important in the culture of many people. In the last twenty years, most of the efforts in Arabic text recognition have been toward the recognition of scanned printed documents [4, 17, 25]. Most of these works have been evaluated on private databases; therefore, the comparison of systems is rather difficult. To our knowledge, there are currently few large-scale image databases of Arabic printed text available for the scientific community. One of the only references we have found is on the ERIM [24] database containing 750 scanned pages collected from Arabic books and magazines. However, it seems difficult to have access to this database. In the Arabic handwriting recognition field, public databases exist such as the freely available IFN/ENIT-database [22]. Open competitions are even regularly organized using this database [19–21].

On the other hand, a corpus is a large structured set of text, electronically stored and processed. A text corpus or lexical database in Arabic is available from different associations or institutes [1, 2, 12]. However, such text corpora are not directly usable for the benchmarking of recognition systems that take images as input. Access to a corpus of both language and images is essential during optical character recognition (OCR) development, particularly while training and testing a recognition application [3]. Excellent corpora have been developed for Latin-based languages, but only a few relate to the Arabic language. This limits the penetration of both corpus linguistics and OCR in Arabic-speaking countries. In [3], the authors describe the construction and provide a comprehensive study and analysis of a multi-modal Arabic corpus (MMAC) that is suitable for use in both OCR development and linguistics. MMAC contains six million Arabic words and includes connected segments as well as naked pieces of Arabic words (NPAWs) and naked words (NWords); a ground truth annotation is offered for each image. MMAC is publicly and freely available.

To bring into account the above information, we initiated the development of a large database of images of printed Arabic words in 2009. This database is used

for our own research and is made available for the scientific community to evaluate their recognition systems. The database has been named Arabic Printed Text Image (APTI).

The purpose of the APTI database is a large-scale benchmarking of open-vocabulary, multi-font, multi-size and multi-style text recognition systems in Arabic. The images in the database are synthetically generated from a large corpus using automated procedures. The challenges that are addressed by the database are in the variability of the sizes, fonts and styles used to generate the images. Special attention is also paid to low resolution images where anti-aliasing is generating noise on the characters to recognize. Naturally, APTI is well suited for the evaluation of screen-based OCR systems that take as input images extracted from screen captures or pdf documents. Performances of classical scanned-based OCR or camera-based OCR systems could also be measured using APTI. However, such evaluations should take into account the absence of typical artefacts present in scanned or camera documents.

Being synthetically generated, the challenges of the database remain multiple:

- Large-scale evaluation with a realistic sampling of most of the Arabic character shapes and their accompanying variations due to ligatures and overlaps.
- Availability of multiple fonts, styles and sizes that must nowadays be treated by recognition systems.
- Emphasis on the low resolution images that are nowadays frequently present on computer screens.
- Isolated word images where inter-word language models cannot be used.
- Semi-blind evaluation protocols with decoupled development/evaluation sets.

Research work on Arabic optical text recognition has increased considerably since the 1980s. Scanner-based OCR has made considerable advances over the two past decades, thanks to the combined progress of the acquisition devices, recognition algorithms and computer capacities. OCR is nowadays practically considered as a solved problem in the case of Latin-based character inputs acquired in high resolution from flat bed scanners. First Arabic OCR systems were made available in the market in the 1990s. Currently, a few commercial systems are available, but the only independent system of comparison was made 15 years ago. Compared to the high quality and widespread usage of OCR systems for Latin characters, Arabic OCR still has to be developed, especially for the case of low resolution printed words.

More challenging tasks are now appearing where the conditions are more adverse, showing a significant drop of the performance in comparison with more classical applications. This is the case for screen-based OCR where inputs are typically at a lower resolution, showing multiple fonts and sizes and including potentially single words with very short sequences of characters. Recognition of low resolution text is quite interesting due to the wide range of applications and occurrence of low resolution text in screen shots, images and videos.

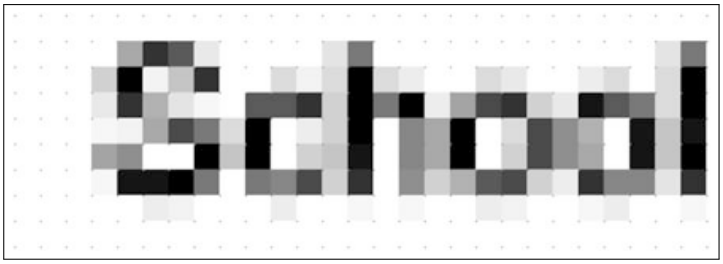

Fig. 18.1 An image of word 'School' at ultra low resolution and anti-aliased

This work is in the field of screen-rendered text OCR applied to the Arabic language. Recognition of screen-rendered text can be used to:

- Recognize low resolution text in videos.
- Provide meanings or translation of text from screen-shots of documents.
- Correct web page errors due to bad background and foreground combination [26].
- Enable web indexing tools to capture semantic important information from web images.
- Develop tools which read screen text for blind or visually impaired people.

The remainder of this chapter is organized as follows. In Sect. 18.2, we illustrate some of the major challenges in the recognition of low resolution text images. The first free APTI database on low resolution will be presented in Sect. 18.3. The first edition of the ICDAR'2011 Arabic recognition competition will be described in Sect. 18.4. In Sect. 18.5, we present the DIVA-REGIM system with results of the competition protocols followed immediately by some conclusions.

18.2 OCR Challenges for Low Resolution Text Images

Recognizing a low resolution text by OCR is a challenging task and involves several difficulties. Screen-rendered text can be on ultra low resolution (see Fig. 18.1) and is generally smoothed to make it look better to the human eye. Smoothed characters are difficult to segment because they are too close to other characters. This, for example, makes contour- and projection-based segmentation inapplicable. Also, the same character of the same logical description (font, size, etc.) is often rendered differently within the same document depending on its position. Furthermore, screen text can be displayed at any screen position. This means that the text can occur at an inhomogeneous background or that single words can be rendered isolated. Baseline detection for isolated words is difficult since commonly used horizontal projections of single words are not sufficient. Generally, the appearance of screen-rendered text depends on the font type, magnification, size, background, position, operating system or application, context and used smoothing algorithm [31].

18.3 APTI Database

Available since July 2009, APTI has been freely distributed to the scientific community for benchmarking purposes.[1] At the time of this writing, more than 20 research groups from universities, research centres, and industries worldwide are working with the APTI database. The APTI database is synthetic, and images are generated using automated procedures. In this section, we present the specificities of this database.

18.3.1 Corpus

APTI is created using a mix of non-decomposable and decomposable Arabic words [5, 7, 9, 13, 16]. Non-decomposable words are formed by country/town/village names, Arabic proper names, general names, Arabic prepositions, etc., whereas decomposable words are generated from root Arabic verbs using Arabic schemes [15]. To generate the lexicon, different Arabic books such as Albukhala of Gahiz[2] and *The Muqaddimah—An introduction to the history* of Ibn Khaldun[3] were parsed. A collection of Arabic newspaper articles were also taken from the Internet as well as a large lexicon file produced by [15]. This parsing procedure totalled 113,284 single different Arabic words, leading to a pretty good coverage of the Arabic words from different disciplines, e.g. literature, culture, art, medicine, and law.

18.3.2 Fonts, Styles and Sizes

Taking as input the Arabic words, the APTI images are generated using 10 different sizes (6, 7, 8, 9, 10, 12, 14, 16, 18 and 24 points) and 10 different fonts as presented in Fig. 18.2. These fonts have been selected to cover different complexities of Arabic printed character shapes, going from simple fonts with no or few overlaps and ligatures like Andalus to more complex fonts rich in overlaps, ligatures and flourishes like Diwani Letter. All word images are generated also using four different styles: plain, italic, bold and a combination of italic and bold. These sizes, fonts and styles are widely used on computer screens, Arabic newspapers and many other

[1] http://diuf.unifr.ch/diva/APTI/

[2] Al-Jahiz (born in Basra, c. 781–December 868 or January 869) was a famous Arab scholar, believed to have been an Afro-Arab of East African descent (http://en.wikipedia.org/wiki/Al-Jahiz).

[3] Ibn Khaldoun (May 27, 1332–March 19, 1406) was a famous historian, scholar, theologian, and statesman born in North Africa in present-day Tunisia (http://en.wikipedia.org/wiki/Ibn_Khaldoun).

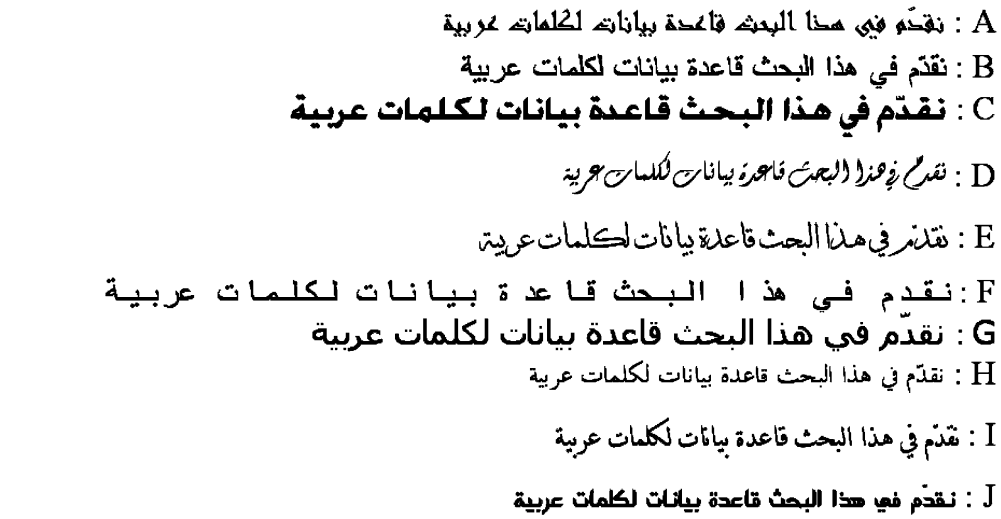

Fig. 18.2 Fonts used to generate the APTI database: (A) Andalus, (B) Arabic Transparent, (C) AdvertisingBold, (D) Diwani Letter, (E) DecoType Thuluth, (F) Simplified Arabic, (G) Tahoma, (H) Traditional Arabic, (I) DecoType Naskh, (J) M Unicode Sara

documents. The combination of fonts, styles and sizes guarantees a wide variability in APTI images.

18.3.3 Procedure for Creating Images

The word images are generated using a developed program. As a consequence, the artefacts or noise usually present on scanned or camera-based documents are not present in the images. Such degradations could actually be artificially added, if needed [6], but it is currently out of the scope of APTI. The text image generation, for example on a screen, can be done in many different ways. They all usually lead to slight variations of the target image. We have opted for a rendering procedure that allows us to include effects of down-sampling and anti-aliasing. These effects are interesting in terms of the variability of the images, especially at low resolution. The procedure involves the down-sampling of a high resolution source image into a low resolution image using anti-aliasing filtering. We also use different grid alignments to introduce variability in the application of the anti-aliasing filter. The details of the procedure are as follows:

1. A grey-scale source image is generated in high resolution (360 pixels/inch) from the current word in the lexicon, using the selected font, size and style.
2. Columns and rows of white pixels are added to the right-hand side and to the top of the image. The number of columns and rows is chosen to have a height and width multiple of the down-sampling factor [28]. This effect allows us to

```
<?xml version="1.0" encoding="UTF-8" ?>
- <wordImage id="78">
  - <content transcription="الآف" nPaws="4">
      <paw id="1" nbChars="1">Alif_I</paw>
      <paw id="2" nbChars="2">Laam_B TildAboveAlif_E</paw>
      <paw id="3" nbChars="2">Laam_B Alif_E</paw>
      <paw id="4" nbChars="1">Faa_I</paw>
    </content>
    <font name="Arabic Transparent" fontStyle="Plain" size="24" />
    <specs encoding="png" width="96" height="36" effect="none" />
    <generation type="downsampling5" renderer="java" filtering="antialiasing" />
  </wordImage>
```

Fig. 18.3 Example of XML file including ground truth information about a given word image

have the same deformation in all images and artificially move the down-sampling grid.
3. Down-sampling and anti-aliasing filtering are applied to obtain the target image in low resolution (72 pixels/inch). The target image is in grey level.

18.3.4 Sources of Variability

APTI presents many sources of variability related to the generation procedure of images. The following list describes some of them:

1. 10 different fonts; 10 different sizes and 4 different styles.
2. Very large vocabulary that allows to test systems on unseen data.
3. Various artefacts of the down-sampling and anti-aliasing filters due to the insertion of columns of white pixels at the beginning and the top of word images.
4. Various forms of ligatures and overlaps of characters due to the large combination of characters in the lexicon and due to the used fonts.
5. Variability of the height of each word image. The height of each word image is related to the sequence of characters appearing in the word.

18.3.5 Ground Truth

In document image analysis and pattern recognition, ground truth refers to the various attributes associated with the text on the image such as the size of tokens, characters, used font, size, etc. Ground truth data is crucial for the training and testing of document image analysis applications [18]. However, each token word image in the APTI database contains ground truth information. Figure 18.3 shows the ground truth XML file containing information about the sequence of characters as well as the generation procedure.

The XML file includes the four following attributes:

Table 18.1 Quantity of words, PAWs and characters in APTI

	Words	PAWs	Characters
	113,284	274,833	648,280
	∗ 10 Fonts ∗ 10 Font Sizes ∗ 4 Font Styles		
Total	45,313,600	109,933,200	259,312,000

1. *Content*: this contains the transcription of Arabic words, the number of pieces of Arabic words (nPaws) and sub-elements for each PAW with the sequence of characters. In our representation, characters are identified using plain English labels as described below.
2. *Font*: this contains the font name, font style and size used to generate the word image.
3. *Specs*: this presents the encoding of image, width, height and eventual additional effect.
4. *Generation*: this indicates the type of generation, the tool used for generation and the used filters in the generation procedure. In the current version of APTI, this element is constant as the same generation procedure has been applied. The type 'downsampling5' is used, indicating that the generation procedure corresponds to a down-sampling, using factor 5, from high resolution source images.

The different character labels can be observed in Table 18.5 showing their statistics through the sets of APTI. As the Arabic character shapes vary according to their position in a word, the character labels also include a suffix to specify the position of the character in the word: B standing for beginning, M for middle, E for end and I for isolated. The character 'Hamza' is always isolated, so we don't use the suffix position for this character. We also artificially inserted characters labels such as 'NuunChadda نّ' or 'YaaChadda يّ' to represent the character shape issued from the combination of 'Nuun ن' and 'Chadda' or 'Yaa ي' and 'Chadda'.

18.3.6 Database Statistics

The APTI database includes 113,284 different single words. Table 18.1 shows the total quantity of word images, PAWs and characters in APTI.

APTI is divided into six equilibrated sets to allow for flexibility in the composition of development and evaluation partitions. Five sets are available for the scientific community, and the sixth one is kept internal for potential evaluation of systems in blind mode (the set 6 was used at the ICDAR'2011 Arabic Recognition Competition presented in Sect. 18.4). The words in each set are different, but the distribution of all used letters is nearly the same in the various sets (see Table 18.5).

Table 18.2 Distribution of letters in set 1 (left) and set 2 (right)

Letter label	Nb. Occ	Isolate	Begin	Middle	End
Alif	15078	ا 5823		ـا 9255	
Baa	4513	ب 128	بـ 1978	ـبـ 2226	ـب 181
Taaa	9926	ت 587	تـ 3626	ـتـ 5332	ـت 381
Thaa	634	ث 12	ثـ 261	ـثـ 341	ـث 20
Jiim	1893	ج 60	جـ 781	ـجـ 1016	ـج 36
Haaa	2953	ح 69	حـ 1135	ـحـ 1648	ـح 101
Xaa	1407	خ 16	خـ 587	ـخـ 782	ـخ 22
Daal	3187	د 988		ـد 2199	
Thaal	514	ذ 167		ـذ 347	
Raa	6304	ر 1813		ـر 4491	
Zaay	1064	ز 389		ـز 675	
Siin	3674	س 68	سـ 1434	ـسـ 2083	ـس 89
Shiin	1457	ش 18	شـ 580	ـشـ 831	ـش 28
Saad	1374	ص 14	صـ 439	ـصـ 882	ـص 39
Daad	922	ض 41	ضـ 358	ـضـ 497	ـض 26
Thaaa	1419	ط 42	طـ 392	ـطـ 920	ـط 65
Taa	242	ظ 6	ظـ 58	ـظـ 163	ـظ 15
Ayn	2764	ع 67	عـ 1003	ـعـ 1575	ـع 119
Ghayn	981	غ 12	غـ 413	ـغـ 543	ـغ 13
Faa	2305	ف 87	فـ 1213	ـفـ 923	ـف 82
Gaaf	2784	ق 97	قـ 937	ـقـ 1614	ـق 136
Kaaf	2101	ك 69	كـ 914	ـكـ 988	ـك 130
Laam	6745	ل 175	لـ 3546	ـلـ 2206	ـل 818
Miim	7871	م 177	مـ 4043	ـمـ 2844	ـم 807
Nuun	7484	ن 2437	نـ 1264	ـنـ 1905	ـن 1878
NuunChadda	225	ّن 0	نّ 0	ـنّـ 225	ـنّ 0
Haa	2670	ه 223	هـ 704	ـهـ 1196	ـه 548
Waaw	4421	و 1621		ـو 2800	
Yaa	6641	ي 317	يـ 2516	ـيـ 2640	ـي 1168
YaaChadda	725	ّي 0	يّ 192	ـيّـ 533	ـيّ 0
Hamza	192	ء 192			
HamzaAboveAlif	1437	أ 1102		ـأ 335	
TaaaClosed	1417	ة 441			ـة 976
HamzaUnderAlif	253	إ 182		ـإ 71	
AlifBroken	162	ى 53			ـى 109
TildAboveAlif	84	آ 32		ـآ 52	
HamzaAboveAlifBroken	210	ئ 3	ئـ 167	ـئـ 39	ـئ 1
HamzaAboveWaaw	89	ؤ 30		ـؤ 59	

Letter label	Nb. Occ	Isolate	Begin	Middle	End
Alif	14925	ا 5777		ـا 9148	
Baa	4763	ب 150	بـ 2039	ـبـ 2344	ـب 230
Taaa	9884	ت 642	تـ 3551	ـتـ 5347	ـت 344
Thaa	633	ث 19	ثـ 230	ـثـ 349	ـث 35
Jiim	1897	ج 54	جـ 756	ـجـ 1034	ـج 53
Haaa	2963	ح 93	حـ 1159	ـحـ 1619	ـح 92
Xaa	1435	خ 18	خـ 622	ـخـ 777	ـخ 18
Daal	3033	د 963		ـد 2070	
Thaal	520	ذ 166		ـذ 354	
Raa	6243	ر 1823		ـر 4420	
Zaay	1054	ز 379		ـز 675	
Siin	3556	س 77	سـ 1338	ـسـ 2041	ـس 100
Shiin	1446	ش 22	شـ 558	ـشـ 838	ـش 28
Saad	1377	ص 22	صـ 420	ـصـ 906	ـص 29
Daad	943	ض 42	ضـ 374	ـضـ 492	ـض 35
Thaaa	1426	ط 38	طـ 401	ـطـ 925	ـط 62
Taa	238	ظ 7	ظـ 66	ـظـ 149	ـظ 16
Ayn	2823	ع 85	عـ 1074	ـعـ 1543	ـع 121
Ghayn	970	غ 15	غـ 444	ـغـ 495	ـغ 16
Faa	2256	ف 62	فـ 1184	ـفـ 937	ـف 73
Gaaf	2734	ق 104	قـ 872	ـقـ 1632	ـق 126
Kaaf	2090	ك 63	كـ 891	ـكـ 1002	ـك 134
Laam	6926	ل 193	لـ 3513	ـلـ 2334	ـل 886
Miim	7836	م 162	مـ 4152	ـمـ 2704	ـم 818
Nuun	7433	ن 2391	نـ 1262	ـنـ 1848	ـن 1932
NuunChadda	224	ّن 0	نّ 0	ـنّـ 224	ـنّ 0
Haa	2687	ه 224	هـ 705	ـهـ 1201	ـه 559
Waaw	4313	و 1480		ـو 2833	
Yaa	6630	ي 317	يـ 2432	ـيـ 2701	ـي 1183
YaaChadda	727	ّي 0	يّ 210	ـيّـ 517	ـيّ 0
Hamza	187	ء 187			
HamzaAboveAlif	1483	أ 1156		ـأ 327	
TaaaClosed	1407	ة 429			ـة 978
HamzaUnderAlif	250	إ 160		ـإ 90	
AlifBroken	161	ى 47			ـى 114
TildAboveAlif	84	آ 40		ـآ 44	
HamzaAboveAlifBroken	208	ئ 0	ئـ 166	ـئـ 34	ـئ 8
HamzaAboveWaaw	90	ؤ 32		ـؤ 58	

18.3.7 Division into Sets

The algorithm for the distribution of words in the different sets has been designed to have similar allocations of letters and words in all sets. This procedure is simply stressing a fair distribution of words that include characters with few occurrences. This type of distribution is important to avoid under-representation of a given character in a given set and therefore to avoid potential problems while training or testing time. Tables 18.2, 18.3, 18.4 present the distribution of each shape of Arabic characters in their respective six sets.

Table 18.3 Distribution of letters in sets 3 and 4 respectively

Letter label	Nb. Occ	Isolate	Begin	Middle	End
Alif	15165	ا 5988		ـا 9177	
Baa	4692	ب 156	بـ 1955	ـبـ 2343	ـب 238
Taaa	9897	ت 617	تـ 3546	ـتـ 5380	ـت 354
Thaa	631	ث 16	ثـ 245	ـثـ 335	ـث 35
Jiim	1887	ج 53	جـ 784	ـجـ 998	ـج 52
Haaa	3017	ح 63	حـ 1194	ـحـ 1659	ـح 101
Xaa	1439	خ 11	خـ 643	ـخـ 765	ـخ 20
Daal	3075	د 947		ـد 2128	
Thaal	528	ذ 185		ـذ 343	
Raa	6169	ر 1746		ـر 4423	
Zaay	1054	ز 362		ـز 692	
Siin	3674	س 75	سـ 1411	ـسـ 2085	ـس 103
Shiin	1418	ش 18	شـ 545	ـشـ 827	ـش 28
Saad	1388	ص 17	صـ 390	ـصـ 948	ـص 33
Daad	936	ض 50	ضـ 346	ـضـ 511	ـض 29
Thaaa	1431	ط 39	طـ 393	ـطـ 937	ـط 62
Taa	240	ظ 1	ظـ 46	ـظـ 176	ـظ 17
Ayn	2769	ع 64	عـ 1015	ـعـ 1560	ـع 130
Ghayn	983	غ 12	غـ 423	ـغـ 530	ـغ 18
Faa	2221	ف 54	فـ 1178	ـفـ 910	ـف 79
Gaaf	2853	ق 107	قـ 984	ـقـ 1640	ـق 122
Kaaf	2099	ك 76	كـ 904	ـكـ 996	ـك 123
Laam	6972	ل 183	لـ 3606	ـلـ 2259	ـل 924
Miim	7957	م 190	مـ 4066	ـمـ 2899	ـم 802
Nuun	7289	ن 2319	نـ 1293	ـنـ 1811	ـن 1866
NuunChadda	224	نّ 0	نّـ 0	ـنّـ 224	ـنّ 0
Haa	2590	ه 192	هـ 631	ـهـ 1222	ـه 546
Waaw	4325	و 1507		ـو 2818	
Yaa	6876	ي 318	يـ 2527	ـيـ 2764	ـي 1270
YaaChadda	709	يّ 0	يّـ 198	ـيّـ 511	ـيّ 0
Hamza	190	ء 190			
HamzaAboveAlif	1455	أ 1133		ـأ 322	
TaaaClosed	1394	ة 435			ـة 959
HamzaUnderAlif	256	إ 169		ـإ 87	
AlifBroken	164	ى 58			ـى 106
TildAboveAlif	83	آ 39		ـآ 44	
HamzaAboveAlifBroken	208	ئ 4	ئـ 170	ـئـ 27	ـئ 7
HamzaAboveWaaw	89	ؤ 21		ـؤ 68	

Letter label	Nb. Occ	Isolate	Begin	Middle	End
Alif	15120	ا 5866		ـا 9254	
Baa	4704	ب 132	بـ 1979	ـبـ 2362	ـب 231
Taaa	9797	ت 633	تـ 3625	ـتـ 5208	ـت 331
Thaa	634	ث 29	ثـ 219	ـثـ 360	ـث 26
Jiim	1924	ج 61	جـ 808	ـجـ 1016	ـج 39
Haaa	2933	ح 68	حـ 1205	ـحـ 1552	ـح 108
Xaa	1401	خ 16	خـ 615	ـخـ 749	ـخ 21
Daal	2990	د 909		ـد 2081	
Thaal	504	ذ 144		ـذ 360	
Raa	6335	ر 1833		ـر 4502	
Zaay	1066	ز 400		ـز 666	
Siin	3512	س 63	سـ 1349	ـسـ 2006	ـس 94
Shiin	1434	ش 17	شـ 596	ـشـ 796	ـش 25
Saad	1411	ص 19	صـ 422	ـصـ 937	ـص 33
Daad	906	ض 34	ضـ 381	ـضـ 457	ـض 34
Thaaa	1426	ط 34	طـ 399	ـطـ 929	ـط 64
Taa	238	ظ 0	ظـ 64	ـظـ 159	ـظ 15
Ayn	2718	ع 72	عـ 1016	ـعـ 1518	ـع 112
Ghayn	984	غ 12	غـ 399	ـغـ 566	ـغ 7
Faa	2313	ف 73	فـ 1264	ـفـ 894	ـف 82
Gaaf	2883	ق 106	قـ 999	ـقـ 1639	ـق 139
Kaaf	2145	ك 86	كـ 935	ـكـ 978	ـك 146
Laam	7002	ل 207	لـ 3656	ـلـ 2247	ـل 892
Miim	7806	م 157	مـ 3963	ـمـ 2848	ـم 838
Nuun	7316	ن 2341	نـ 1239	ـنـ 1860	ـن 1876
NuunChadda	223	نّ 0	نّـ 0	ـنّـ 223	ـنّ 0
Haa	2718	ه 201	هـ 681	ـهـ 1252	ـه 585
Waaw	4333	و 1494		ـو 2839	
Yaa	6685	ي 322	يـ 2443	ـيـ 2699	ـي 1221
YaaChadda	719	يّ 0	يّـ 215	ـيّـ 504	ـيّ 0
Hamza	193	ء 193			
HamzaAboveAlif	1512	أ 1164		ـأ 348	
TaaaClosed	1364	ة 398			ـة 966
HamzaUnderAlif	247	إ 171		ـإ 76	
AlifBroken	163	ى 42			ـى 121
TildAboveAlif	83	آ 38		ـآ 45	
HamzaAboveAlifBroken	209	ئ 5	ئـ 161	ـئـ 35	ـئ 8
HamzaAboveWaaw	91	ؤ 24		ـؤ 67	

18.3.8 APTI Evaluation Protocols

In this section, we propose the definition of a set of robust benchmarking protocols on top of the APTI database. Preliminary experiments with a baseline recognition system have helped in calibrating and validating these protocols. From the obtained results, we believe that the large amount of data available in APTI and the different sources of variability (cf. Sect. 18.3.4) make it well suited for significant and challenging system evaluation.

Table 18.4 Distribution of letters in sets 5 and 6 respectively

Letter label	Nb. Occ	Isolate	Begin	Middle	End
Alif	15046	ا 5689		ـا 9357	
Baa	4730	ب 161	بـ 1991	ـبـ 2341	ـب 237
Taaa	9942	ت 580	تـ 3629	ـتـ 5389	ـت 344
Thaa	643	ث 26	ثـ 242	ـثـ 347	ـث 28
Jiim	1915	ج 60	جـ 809	ـجـ 990	ـج 56
Haaa	3000	ح 83	حـ 1134	ـحـ 1680	ـح 103
Xaa	1403	خ 15	خـ 611	ـخـ 754	ـخ 23
Daal	3028	د 901		ـد 2127	
Thaal	516	ذ 159		ـذ 357	
Raa	6253	ر 1824		ـر 4429	
Zaay	1042	ز 386		ـز 656	
Siin	3629	س 59	سـ 1401	ـسـ 2091	ـس 78
Shiin	1455	ش 25	شـ 566	ـشـ 838	ـش 26
Saad	1371	ص 14	صـ 413	ـصـ 896	ـص 48
Daad	921	ض 41	ضـ 369	ـضـ 470	ـض 41
Thaaa	1446	ط 33	طـ 412	ـطـ 934	ـط 67
Taa	239	ظ 5	ظـ 52	ـظـ 169	ـظ 13
Ayn	2755	ع 68	عـ 1017	ـعـ 1552	ـع 118
Ghayn	990	غ 15	غـ 422	ـغـ 534	ـغ 19
Faa	2339	ف 73	فـ 1257	ـفـ 920	ـف 89
Gaaf	2762	ق 103	قـ 959	ـقـ 1574	ـق 126
Kaaf	2136	ك 84	كـ 914	ـكـ 980	ـك 158
Laam	6790	ل 188	لـ 3433	ـلـ 2288	ـل 881
Miim	7797	م 175	مـ 4067	ـمـ 2732	ـم 823
Nuun	7400	ن 2435	نـ 1273	ـنـ 1825	ـن 1867
NuunChadda	224	ںّ 0	ںّـ 0	ـںّـ 224	ـںّ 0
Haa	2705	ه 178	هـ 699	ـهـ 1297	ـه 531
Waaw	4264	و 1466		ـو 2798	
Yaa	6648	ي 327	يـ 2507	ـيـ 2656	ـي 1160
YaaChadda	735	يّ 0	يّـ 168	ـيّـ 567	ـيّ 0
Hamza	192	ء 192			
HamzaAboveAlif	1456	أ 1158		ـأ 298	
TaaaClosed	1409	ة 433			ـة 976
HamzaUnderAlif	248	إ 171		ـإ 77	
AlifBroken	161	ى 55			ـى 106
TildAboveAlif	83	آ 46		ـآ 37	
HamzaAboveAlifBroken	208	ئ 2	ئـ 167	ـئـ 32	ـئ 7
HamzaAboveWaaw	89	ؤ 28		ـؤ 61	

Letter label	Nb. Occ	Isolate	Begin	Middle	End
Alif	15019	ا 5797		ـا 9222	
Baa	4717	ب 146	بـ 1998	ـبـ 2354	ـب 219
Taaa	9897	ت 641	تـ 3612	ـتـ 5304	ـت 340
Thaa	628	ث 22	ثـ 227	ـثـ 353	ـث 26
Jiim	1939	ج 49	جـ 803	ـجـ 1048	ـج 39
Haaa	3000	ح 83	حـ 1180	ـحـ 1655	ـح 82
Xaa	1407	خ 7	خـ 618	ـخـ 751	ـخ 31
Daal	3086	د 939		ـد 2147	
Thaal	518	ذ 164		ـذ 354	
Raa	6267	ر 1864		ـر 4403	
Zaay	1045	ز 377		ـز 668	
Siin	3603	س 73	سـ 1359	ـسـ 2062	ـس 109
Shiin	1458	ش 26	شـ 582	ـشـ 817	ـش 33
Saad	1389	ص 21	صـ 415	ـصـ 921	ـص 32
Daad	920	ض 43	ضـ 335	ـضـ 503	ـض 39
Thaaa	1462	ط 24	طـ 428	ـطـ 937	ـط 73
Taa	241	ظ 4	ظـ 65	ـظـ 158	ـظ 14
Ayn	2723	ع 80	عـ 1007	ـعـ 1510	ـع 126
Ghayn	1004	غ 15	غـ 425	ـغـ 540	ـغ 24
Faa	2315	ف 62	فـ 1226	ـفـ 928	ـف 99
Gaaf	2803	ق 99	قـ 974	ـقـ 1584	ـق 146
Kaaf	2140	ك 85	كـ 913	ـكـ 1004	ـك 138
Laam	6724	ل 174	لـ 3466	ـلـ 2203	ـل 881
Miim	7817	م 166	مـ 4038	ـمـ 2779	ـم 834
Nuun	7264	ن 2411	نـ 1231	ـنـ 1835	ـن 1787
NuunChadda	223	ںّ 0	ںّـ 0	ـںّـ 223	ـںّ 0
Haa	2724	ه 230	هـ 695	ـهـ 1236	ـه 565
Waaw	4352	و 1514		ـو 2838	
Yaa	6735	ي 301	يـ 2535	ـيـ 2652	ـي 1250
YaaChadda	733	يّ	يّـ 199	ـيّـ 534	ـيّ
Hamza	188	ء 188			
HamzaAboveAlif	1427	أ 1113		ـأ 314	
TaaaClosed	1385	ة 430			ـة 955
HamzaUnderAlif	247	إ 179		ـإ 68	
AlifBroken	161	ى 43			ـى 118
TildAboveAlif	83	آ 37		ـآ 46	
HamzaAboveAlifBroken	210	ئ 6	ئـ 164	ـئـ 39	ـئ 1
HamzaAboveWaaw	90	ؤ 23		ـؤ 67	

Error Estimation

The objective of any benchmarking of recognition systems is to estimate, as reliably as possible, the classification error rate $\hat{P}_e$. It is important to remember that, whatever the task and data used, $\hat{P}_e$ is a function of the split of the data into training and test sets. Different splits will result in different error estimates. APTI is composed of quite large sets of data, which helps in reaching stable estimates of $\hat{P}_e$. Our objective is then to obtain a reliable estimate of $\hat{P}_e$ while keeping the computation load tractable. Therefore, we have opted for a *rotation method*, as described in [14, Sect. 7]. The idea is to reach a trade-off between the *holdout method*, which leads to pessimistic and biased values of the error rate, and the *leave-one-out method*, which

Fig. 18.4 Illustration of the rotation method. For a given partition, the training sets are depicted in *dark grey* and the testing sets in *light grey*

gives a better estimate but at the cost of larger computational requirements. The rotation method we are proposing is illustrated in Fig. 18.4. The procedure is to perform independent runs on five different partitions between training and testing data.

The final error estimate is taken as the average of the error rates obtained on the different partitions:

$$\hat{P}_e = \frac{1}{5}\sum_{i=1}^{5} \hat{P}_{e,i} \tag{18.1}$$

In the previous equation, $\hat{P}_{e,i}$ is the error rate obtained independently on a trained and tested system using the sets defined in partition i. The procedure actually corresponds to computing the average performance of five independent systems.

Training and Testing Conditions

Using the procedure described in Sect. 18.3.8, we can define different combinations of training and testing conditions. The objectives are to measure the impact of some of the variability of the data. We therefore propose 20 protocols as summarized in [28].

18.4 ICDAR'2011 Arabic Recognition Competition

This competition was organized by the Document, Image and Voice Analysis (DIVA) research group from the University of Fribourg, Switzerland in collaboration with the REsearch Group on Intelligent Machines (REGIM) group at Ecole Nationale d'Ingénieurs de Sfax (ENIS), from the University of Sfax, Tunisia and the group at the Institute of Communications Technology (IFN) of the Technical University of Braunschweig, Germany. The competition was organized in a 'blind' manner. The testing data for the evaluation is composed of an unpublished set (*set 6* of APTI) which is kept secret for evaluation purposes. The participants were asked to send an executable version of their recognizer to the organizers who, in turn, arrange to run the systems against an unseen set of data. We invited groups working on Arabic word recognition to adapt their system to the APTI database and send us executables of their systems. The scientific objectives of this first edition are to measure the impact of font size on the recognition performances. This is evaluated

Table 18.5 Distribution of characters in the different sets

Char label (Char)	Set 1	Set 2	Set 3	Set 4	Set 5	Set 6
Alif (ا)	15,078	14,925	15,165	15,120	15,046	15,019
Baa (ب)	4513	4763	4692	4704	4730	4717
Taaa (ت)	9926	9884	9897	9797	9942	9897
Thaa (ث)	634	633	631	634	643	628
Jiim (ج)	1893	1897	1887	1924	1915	1939
Haaa (ح)	2953	2963	3017	2933	3000	3000
Xaa (خ)	1407	1435	1439	1401	1403	1407
Daal (د)	3187	3033	3075	2990	3028	3086
Thaal (ذ)	514	520	528	504	516	518
Raa (ر)	6304	6243	6169	6335	6253	6267
Zaay (ز)	1064	1054	1054	1066	1042	1045
Siin (س)	3674	3556	3674	3512	3629	3603
Shiin (ش)	1457	1446	1418	1434	1455	1458
Saad (ص)	1374	1377	1388	1411	1371	1389
Daad (ض)	922	943	936	906	921	920
Thaaa (ط)	1419	1426	1431	1426	1446	1462
Taa (ظ)	242	238	240	238	239	241
Ayn (ع)	2764	2823	2769	2718	2755	2723
Ghayn (غ)	981	970	983	984	990	1004
Faa (ف)	2305	2256	2221	2313	2339	2315
Gaaf (ق)	2784	2734	2853	2883	2762	2803
Kaaf (ك)	2101	2090	2099	2145	2136	2140
Laam (ل)	6745	6926	6972	7002	6790	6724
Miim (م)	7871	7836	7957	7806	7797	7817
Nuun (ن)	7484	7433	7289	7316	7400	7264
Haa (ه)	2670	2687	2590	2718	2705	2724
Waaw (و)	4421	4313	4325	4333	4264	4352
Yaa (ي)	6641	6630	6876	6685	6648	6735
NuunChadda (نّ)	225	224	224	223	224	223
YaaChadda (يّ)	725	727	709	719	735	733
Hamza (ء)	192	187	190	193	192	188
HamzaAboveAlif (أَ)	1437	1483	1455	1512	1456	1427
HamzaUnderAlif (إِ)	253	250	256	247	248	247

Table 18.5 (Continued)

Char label (Char)	Set 1	Set 2	Set 3	Set 4	Set 5	Set 6
TildAboveAlif (آ)	84	84	83	83	83	83
TaaaClosed (ة)	1417	1407	1394	1364	1409	1385
AlifBroken (ى)	162	161	164	163	161	161
HamzaAboveAlifBroken (ىء)	210	208	208	209	208	210
HamzaAboveWaaw (وء)	89	90	89	91	89	90
Quantity of characters	108,122	107,855	108,347	108,042	107,970	107,944
Quantity of PAWs	45,982	45,740	45,792	45,884	45,630	45,805
Quantity of words	18,897	18,892	18,886	18,875	18,868	18,866

in mono-font and multi-font contexts. The protocols are defined to evaluate the capacity of the recognition systems to handle different sizes and fonts using digitally low resolution images in order find a robust approach to screen-based OCR.

The evaluation was reported as word and character recognition rates. In this first edition of the competition, we have proposed two protocols, as described below.

18.4.1 Mono-font Competition Protocol—First APTI Protocol for Competition: APTIPC1

In this protocol, we test Arabic mono-font and multi-size systems trained on the Arabic Transparent font and sizes from 6 to 24.

- *Tested Fonts*: Arabic Transparent.
- *Tested Style*: Plain.
- *Tested Sizes*: 6, 8, 10, 12, 18, 24.
- *Set 6 word images*: 18,866 for each size/font.
- *Number of tests in APTIPC1*: 6.

18.4.2 Multi-font Competition Protocol—Second APTI Protocol for Competition: APTIPC2

In this protocol, we test Arabic multi-font and multi-size systems trained on five fonts and sizes from 6 to 24.

- *Tested Fonts*: Diwani Letter, Andalus, Arabic Transparent, Simplified Arabic and Traditional Arabic.
- *Tested Style*: Plain.

- *Tested Sizes*: 6, 8, 10, 12, 18, 24.
- *Set 6 word images*: 18,866 for each size/font.
- *Number of tests in APTIPC2*: 30.

18.4.3 Participating Systems

The following section gives a short description of the systems submitted to the ICDAR'2011 Arabic Recognition Competition: Multi-Font Multi-Size Digitally Represented Text. The system descriptions vary in length according to the level of detail provided by the participants.

IPSAR System

The IPSAR system was submitted by Samir Ouis, Mohammad S. Khorsheed and Khalid Alfaifi, members of the Image Processing and Signal Analysis & Recognition (IPSAR) Group. This group is part of the Computer Research Institute (CRI) at King Abdulaziz City for Science & Technology (KACST) from the kingdom of Saudi Arabia.

IPSARec is a cursive Arabic script recognition system where ligatures, overlaps and style variation pose challenges to the recognition system. It is based on the Hidden Markov Model Toolkit (HTK), a portable toolkit for speech recognition systems which is customized here to recognize characters. IPSARec is an omnifont, unlimited vocabulary recognition system. It does not require segmentation. The proposed system proceeds with three main stages: extracting a set of features from the input images, clustering the feature set according to a pre-defined codebook and finally, recognizing the characters.

Each word/line image is transferred into a sequence of feature vectors. Those features are extracted from overlapping vertical windows, divided into cells where each cell includes a predefined number of pixels, along the word/line image, then clustered into discrete symbols.

Stage two is performed within HTK. It couples the feature vectors with the corresponding ground truth to estimate the character model parameters. The final output of this stage is a lexicon-free system to recognize cursive Arabic text. During recognition, an input pattern of discrete symbols representing the word/line image is injected to the global model which outputs a stream of characters matching the text line.

For more details about this system, we refer to [17].

UPV-BHMM Systems

These systems were submitted by Ihab Alkhoury, Adria Gimenez and Alfons Juan, from the Universitat Politecnica de Valencia (UPV), Spain. They are based

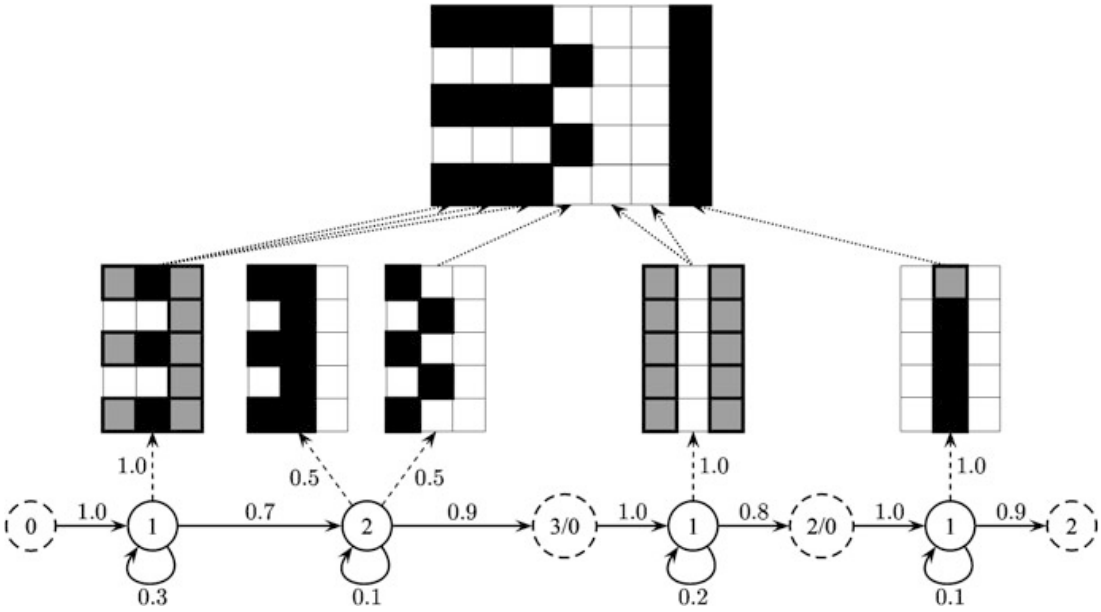

Fig. 18.5 Generation of a 7×5 word image of the number 31 from a sequence of 3 windowed ($W = 3$) BHMMs for the characters 3, 'space' and 1

on Bernoulli HMMs (BHMMs), that is, HMMs in which conventional Gaussian mixture density functions are replaced with Bernoulli mixture probability functions [10]. Also, in contrast to the basic approach followed in [10], in which narrow, one-column slices of binary pixels are fed into BHMMs, the UPV-BHMM systems are based on a sliding window of adequate width to better capture image context at each horizontal position of the word image. This new, windowed version of the basic approach is described in [11]. As an example, Fig. 18.5 shows the generation of a 7×5 word image of the number 31 from a sequence of 3 windowed ($W = 3$) BHMMs for the characters 3, 'space' and 1.

The UPV-PRHLT systems were trained from input images scaled in height to 40 pixels (while keeping the aspect ratio) after adding a certain number of white pixel rows to both top and bottom sides of each image, and then binarized with the Otsu algorithm. A sliding window of width 9 was applied, and thus the resulting input (binary) feature vectors for the BHMMs had 360 bits. The number of states per character was adjusted to 5 states for images with font size of 6, and 6 states for other font sizes. Similarly, the number of mixture components per state was empirically adjusted to 64. The estimation and recognition parameters were carried out using the expectation maximization (EM) algorithm.

Two systems were submitted: *UPV-PRHLT-REC1* and *UPV-PRHLT-REC2*. They are used for both tasks/protocols. In the first task (one style), there are no differences between systems; one model for each font size is trained and used later to recognize the test corpus. For the second task, in the first system, for each font size, a different model for each font style is trained. The test corpus is recognized on all models, and the recognized text word of the highest probability is selected. For the second task in the other system, a different character is considered for each style. A model for all styles together is trained and used to recognize the test corpus.

Table 18.6 APTIPC1—ICDAR'2011 competition results for participant systems

System		Size						Mean RR
		6	8	10	12	18	24	
IPSAR System	WRR	5.7	73.3	75.0	83.1	77.1	77.5	*65.3*
	CRR	59.4	94.2	95.1	96.9	95.7	**96.8**	*89.7*
UPV-PRHLT-REC1	WRR	**94.5**	**97.4**	**96.7**	**92.5**	**84.6**	**84.4**	**91.7**
	CRR	**99.0**	**99.6**	**99.4**	**98.7**	**96.9**	96.0	**98.3**
UPV-PRHLT-REC2	WRR	94.5	97.4	96.7	92.5	84.6	84.4	*91.7*
	CRR	99.0	99.6	99.4	98.7	96.9	96.0	*98.3*

18.4.4 Competition Results

All systems have been tested using the *set 6* (18,866 single word images) of the APTI database in different sizes and fonts. All participants sent us a running version of their recognition systems. The systems can be classified into two classes depending on the operating system: two systems are developed under Linux (UPV-PRHLT-REC1 and UPV-PRHLT-REC2) and one system under the Microsoft Windows environment.

Table 18.6 presents all system results of the first APTI protocol (APTIPC1). For each test the best result is marked in bold.

This first test is mono-font and mono-size. The test images presented to the systems are those using the font Arabic Transparent, plain and sizes 6, 8, 10, 12, 18 and 24. For most of the systems, we observed good results in character recognition and slightly worse results for word recognition. Both UPV-BHMM systems have the same behaviour and show the best results with an average of 91.7 % for the word recognition rate and 98.3 % for the character recognition rate. Compared to other competition systems, the IPSAR system has the best character recognition rate on size 24.

Tables 18.7, 18.8 and 18.9 present system results of the second APTI protocol (APTIPC2) for competition. This second test is multi-font and mono-size. The test images presented to the systems are those using the fonts (Arabic Transparent, Andalus, Simplified Arabic, Traditional Arabic and Diwani Letter), plain and sizes 6, 8, 10, 12, 18 and 24.

In APTIPC2, the recognition rate is not as good as in APTIPC1 for the Arabic Transparent font. The best system is UPV-PRHLT-REC1 with an average of 83.4 % for the word recognition rate and 96.4 % for the character recognition rate.

The UPV-PRHLT-REC1 system shares good results for most fonts and sizes in this APTIPC2. The IPSAR system gives good results for the Traditional Arabic and Diwani Letter fonts in font size 10, 12 and 24.

Table 18.7 APTIPC2—IPSAR system results

Font		Size						Mean RR
		6	8	10	12	18	24	
Andalus	WRR	13.9	35.7	65.6	73.8	69.5	64.5	53.8
	CRR	67.4	82.4	92.4	94.4	93.0	92.5	87.0
Arabic Transparent	WRR	29.9	40.0	73.2	74.9	65.9	69.1	58.8
	CRR	78.2	84.4	94.1	95.1	93.9	95.5	90.2
Simplified Arabic	WRR	30.8	39.8	73.2	75.5	66.2	68.6	59.0
	CRR	77.6	84.3	94.2	94.9	93.1	94.4	89.8
Traditional Arabic	WRR	4.6	3.4	**46.7**	**55.1**	**52.9**	**50.4**	35.5
	CRR	49.8	49.2	**85.9**	**88.5**	**87.5**	**88.3**	74.9
Diwani Letter	WRR	9.7	3.3	**39.9**	**55.8**	49.5	**64.0**	37.0
	CRR	60.1	48.3	**83.4**	**89.1**	91.7	**92.6**	77.5
Mean RR of the system							WRR	48.8
							CRR	83.9

Table 18.8 APTIPC2—UPV-PRHLT-REC1 system results

Font		Size						Mean RR
		6	8	10	12	18	24	
Andalus	WRR	**94.1**	**75.5**	**81.1**	**83.6**	**83.9**	**85.0**	**83.8**
	CRR	**98.9**	**94.8**	**96.1**	**96.7**	**96.7**	**97.0**	**96.7**
Arabic Transparent	WRR	**94.7**	78.2	78.9	**81.8**	**83.1**	**83.8**	**83.4**
	CRR	**99.0**	95.2	95.5	96.1	**96.2**	**96.1**	**96.4**
Simplified Arabic	WRR	**95.8**	82.4	**84.2**	**85.3**	**85.6**	**88.0**	**86.9**
	CRR	**99.2**	96.2	**96.7**	**96.9**	**97.0**	**97.4**	**97.2**
Traditional Arabic	WRR	**57.6**	**38.3**	43.6	43.5	42.9	46.2	**45.4**
	CRR	**89.3**	**81.9**	84.3	83.6	83.5	85.0	**84.6**
Diwani Letter	WRR	**61.7**	**27.7**	30.9	31.6	**76.4**	35.1	**43.9**
	CRR	**90.9**	**75.8**	77.8	78.1	**94.9**	79.6	**82.8**
Mean RR of the system							WRR	68.7
							CRR	91.5

18.5 DIVA-REGIM System

The DIVA-REGIM system is part of a joint collaboration between the DIVA (Document, Image and Voice Analysis) group from the University of Fribourg, Switzerland and the REGIM (REsearch Group on Intelligent Machines) group from the Uni-

Table 18.9 APTIPC2—UPV-PRHLT-REC2 system results

Font		Size						Mean RR
		6	8	10	12	18	24	
Andalus	WRR	83.1	73.6	79.5	77.7	71.1	71.7	76.1
	CRR	96.0	94.1	95.1	94.9	93.6	93.5	94.5
Arabic Transparent	WRR	86.1	**84.3**	**84.1**	81.1	75.5	75.6	81.1
	CRR	97.1	**96.5**	**96.6**	**96.1**	94.9	94.8	96.0
Simplified Arabic	WRR	87.6	**82.6**	83.5	81.2	74.2	76.2	80.9
	CRR	97.4	**96.1**	96.5	96.1	94.7	95.0	96.0
Traditional Arabic	WRR	43.7	36.9	42.3	40.9	37.6	40.2	40.2
	CRR	83.6	80.5	83.2	82.1	80.8	82.2	82.1
Diwani Letter	WRR	41.9	26.4	29.7	29.2	68.4	29.9	37.6
	CRR	83.2	74.5	76.8	76.5	93.4	76.7	80.2
Mean RR of the system							WRR	63.2
							CRR	89.7

versity of Sfax, Tunisia. This system is a cascading system working in three steps: feature extraction, font recognition and word recognition using font-dependent models.

18.5.1 Pre-processing

The pre-processing phase aims at the reduction of the variability between character shapes due to misalignment on the Y-axis. Classically, this pre-processing phase normalizes all inputs by shifting the images so that the characters of a word or sequence of words are aligned vertically according to a common baseline.

A data-driven baseline detection system is proposed in this work. The idea is to detect a probable baseline region using data-driven methods trained on local character features. Once a probable baseline region is recognized by the Gaussian mixture models (GMMs)-based system, the final position of the baseline is fine-tuned using the classical horizontal projection histogram, but limited to this region. The baseline recognition system is actually similar to the system presented in [30] for Arabic font recognition. Each word image is normalized in grey level into a rectangle with fixed height and then transformed into a sequence of feature vectors computed from a narrow analysis window, sliding from right to left on the word image. Again, the features used here for the baseline detection are actually the same as for the font recognition system and are presented in [30]. In our settings, the analysis window is shifted by 1 pixel for each feature vector. We performed several tests to determine the optimal size of the window and we converged to a 4 pixel width and 30 pixel height.

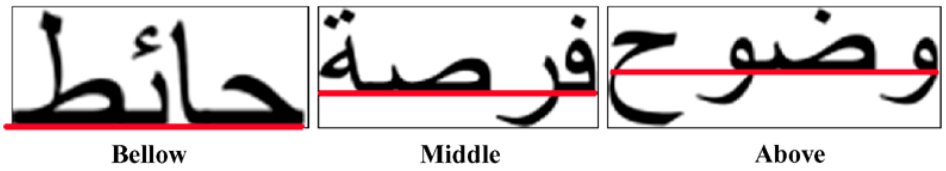

Fig. 18.6 Example of three baseline positions considering the bounding box of single word images

GMMs are used to estimate the likelihoods of three baseline positions (called here *below*, *middle* and *above* positions) as illustrated in Fig. 18.6. Each position is represented by a single GMM, which can actually be seen as a single-state HMM. Assuming the independence of the feature vectors, the GMMs are able to compute a global likelihood of a baseline position simply by multiplying the local likelihoods of each feature vectors computed separately. Each model is trained using an expectation maximization procedure by pooling a large quantity of feature vectors from words in known baseline positions [8]. Being state-less, the obtained models are currently independent of any character but become conditioned to the three baseline positions. Thanks to the large quantity of data, models can typically scale up to a large number of Gaussians (in our settings 8192 Gaussians). At testing time, the GMM showing the largest likelihood is selected, indicating a probable baseline region. Finally, the baseline position is fine-tuned by computing horizontal projection histograms limited to the recognized region.

18.5.2 Feature Extraction

The proposed feature extraction works on binary and grey level images. It depends on horizontal sliding window and vertical frames for a word with specific Arabic fonts (each horizontal window divided into vertical frames without overlap). The width of the horizontal sliding window could be w pixels, where w is an integer number that is determined empirically depending on the developed system for each Arabic font. The height of this window is equal to h pixels, where h represents a fixed integer number. The narrow analysis window slides horizontally from right to left on the word image with a shift of s pixels, where s is an integer window equal to 1. This allows us to take enough samples to be able to reliably estimate character models. For complex Arabic fonts, each horizontal frame is divided into cells where the cell height (C_h) is fixed. This yields a fixed number of cells in each frame according to the normalized word image height. Figure 18.7 illustrates the basic definitions used above.

In our case, the analysis window has a uniform size and moves one pixel from right to left. We conducted several tests to determine the optimal size of the sliding window according to the Arabic font used. As a result, no segmentation into letters is made, and the word image is transformed into a matrix of values where the number of lines corresponds to the number of analysis windows, and the number of columns is equal to the number of coefficients in each feature vector. The feature extraction is divided into two parts. The first part extracts, for each window:

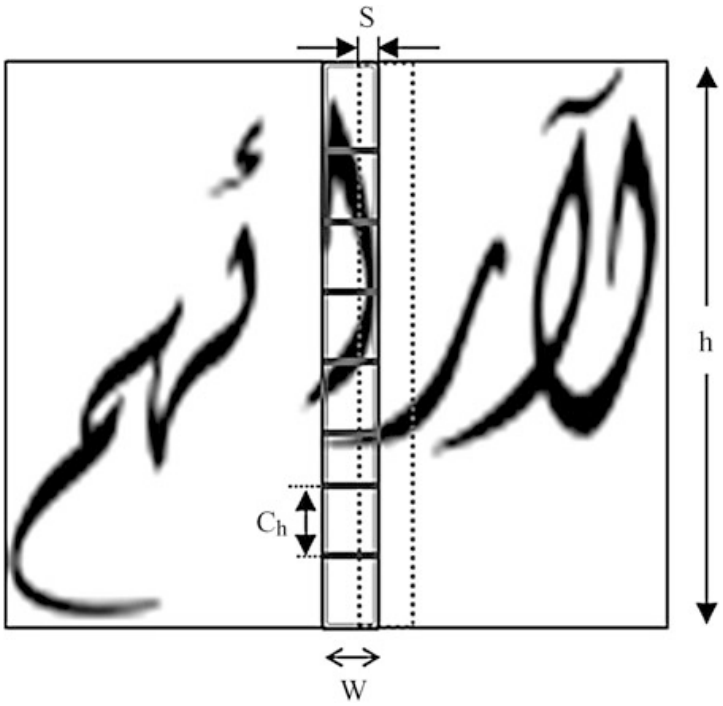

Fig. 18.7 Basic definition used in sliding window feature extraction

- Number $N1$ of black connected components.
- Number $N2$ of white connected components.
- Ratio $N1/N2$.
- Position of the smallest black connected component divided by the height of the window.
- Sum of the perimeter P of all components in window/perimeter of window P_w.
- Compactness $(4\pi A)p^2$ where P is the shape perimeter in window and A is the area.
- Gravity centre of the window, of the right and left half and of the first third, the second and the last part of the window: $\sum_{i=1}^{n} \frac{x_i}{nW}$; $\sum_{i=1}^{n} \frac{y_i}{nH}$ where W is the width and H is the height of the window.
- Position of baseline/height image.
- Number of extremum in vertical projection.
- Number of extremum in horizontal projection.
- Size of the smallest connected component.
- Density of black pixels in the window.
- Density of black pixels below the low baseline.
- Density of black pixels above the low baseline.
- Densities of black pixels in each column of the window. As the width of the window is w pixels, it has w columns in each window.
  When $n(i)$ is the number of black pixels in the cell i, and $b(i)$ is the intensity of the cell i:

$$\begin{cases} b(i) = 0 & \text{if } n(i) = 0 \\ b(i) = 1 & \text{else} \end{cases}$$

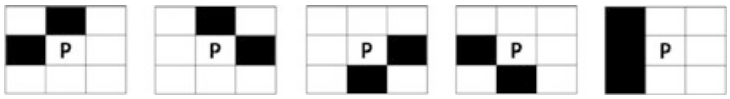

Fig. 18.8 Five types of concavity configurations for a background pixel P

- Number of black/white transitions between cells: $f = \sum_{i=2}^{n_c} |b(i) - b(i-1)|$ where n_c is the number of cells in the window.
- Number of black/white transitions between cells located above the low baseline.
- Position difference between the gravity centres g of writing pixels in two consecutive windows: $f = g(t) - g(t-1)$.
- Area belonging to the text gravity centre in the window (up area $f = 1$, medium area $f = 2$, below area $f = 3$).
- Number of white pixels that belong to one of the five configurations shown in Fig. 18.8. The number of pixels in each configuration is then normalized by the number of pixels in the window.
- Number of background pixels in the five configurations mentioned above but only for pixels located in the middle area of writing, between the two baselines (see Fig. 18.9).
- Number of background pixels in the five configurations mentioned above but only for pixels located in the lower area of writing, below the lower baseline.
- Number of background pixels in the five configurations mentioned above but only for pixels located in the upper area of writing, over the upper baseline.
- The moment invariants (7 moments).
- The affine moment invariants (6 moments).
- The Zernike moments (12 moments).
- The Fourier descriptors (9 descriptors).
- The histogram of the Freeman directions (8 directions).
- The sum of the gradient norms.

The different recognition systems that depend on the font do not use the same feature. For all systems, however, each feature vector x_n has M components including $M/2$ basis features concatenated with $M/2$ delta coefficients computed as a linear difference of the basis features in adjacent windows. The deltas are computed in a similar way as in speech recognition, to include larger contextual information in an analysis window using the following formula:

$$\begin{cases} \Delta x_n^j = x_{n+1}^j - x_{n-1}^j, & \forall 1 < j < M/2 \\ \Delta x_n^j = x_n & \text{where } n = 0 \text{ or } n = N \end{cases}$$

18.5.3 Character Models Training

First, starting from all Arabic character shapes (more than 120), we grouped similar character shapes into 65 models according to the following rules: (1) beginning and middle shapes share the same model; (2) end and isolated shapes share the same

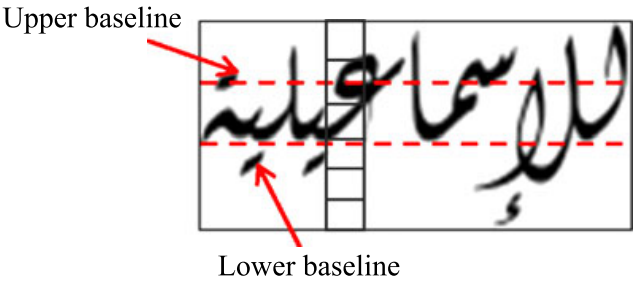

Fig. 18.9 Upper and lower baselines on sample data

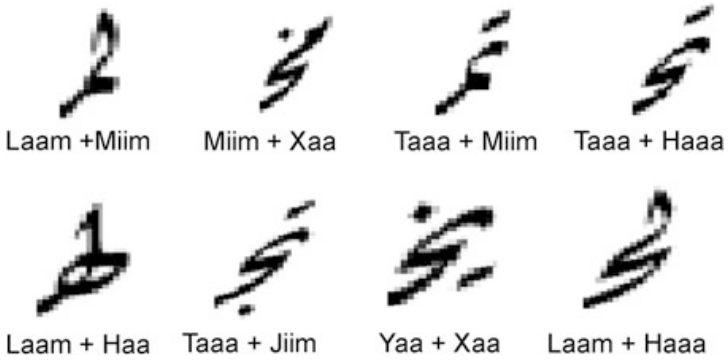

Fig. 18.10 Additional sub-model examples for *Diwani Letter* font

model. These rules apply for all characters with the exception of the characters *Ayn* 'ع' and *ghayn* 'غ' where the beginning, middle, end and isolated shapes are very different. This strategy of grouping is natural as beginning-middle and end-isolated character shapes are visually similar. The selection procedure of the different sub-models has been driven by grouping shapes of letters presenting few variations. The grouping strategy is explained in more detail in [27, 29]. Our hypothesis here is that the emission probability estimators based on Gaussian mixtures will offer enough flexibility to model the common parts and the variations within each letter category. Using the terminology introduced for speech recognition [23], our models are said to be context independent; i.e., each sub-model is considered independent from the next.

Second, for the used fonts presenting many ligatures between letters, we have added a new character sub-model: a selected set of their corresponding variations. Figure 18.10 presents some examples.

18.5.4 Ergodic Topology

In this topology every sub-model can be reached from every other sub-model. All transitions from one sub-model to another are allowed. Using ergodic topology offers the advantage of relatively lightweight memory and CPU footprint, when compared to more heavyweight approaches based on finite-state or stochastic grammars.

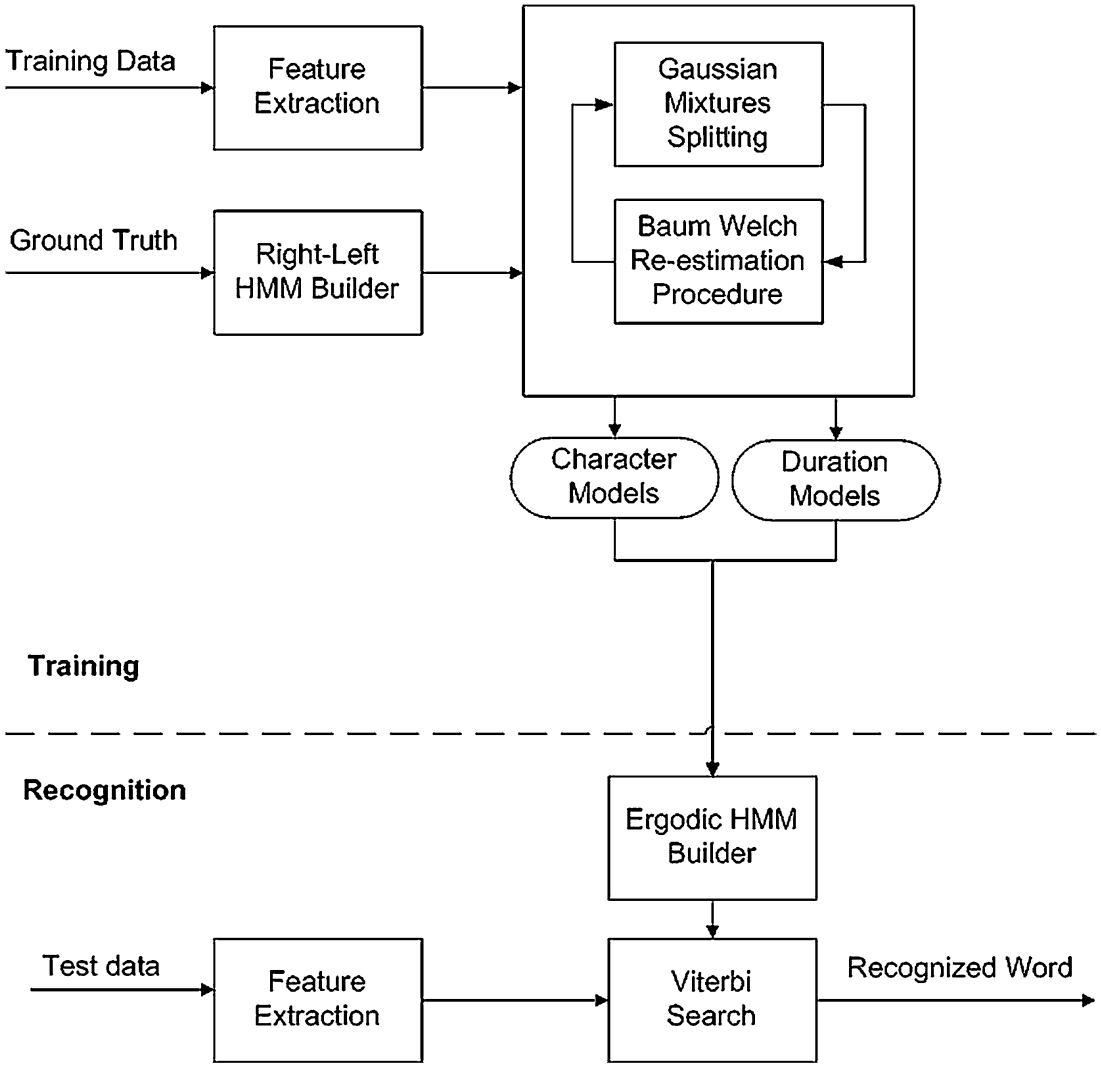

Fig. 18.11 HMM-based word recognition system

18.5.5 Training and Recognition

Our word recognition system is based on hidden Markov models (HMMs). The DIVA-REGIM system has a similar architecture to the one presented in [27]. One of its main characteristics is that it is open vocabulary, i.e. able to recognize any Arabic printed word on ultra low resolution. The training-testing system architecture is illustrated in Fig. 18.11. Note that the baseline detection system shares a similar training-testing architecture; the only difference is the fact that HMMs are here used instead of GMMs.

We used the Hidden Markov Model Toolkit (HTK) to realize our evaluation [32]. HTK was originally developed at the Speech Vision and Robotics Group of the Cambridge University Engineering Department (CUED). This toolbox has been built to experiment with HMMs and has been extensively used in speech recognition research. HTK is a set of command line executables used for initializing, modify-

Table 18.10 APTIPC1—DIVA-REGIM system results

System		Size						Mean RR
		6	8	10	12	18	24	
DIVA-REGIM	WRR	97.47	98.67	99.02	99.32	99.44	99.76	98.95
	CRR	99.74	99.82	99.86	99.92	99.94	99.97	99.88

ing, training and testing HMMs. The use of HTK typically goes through four phases: preparation of data, training, recognition and recognition performance evaluation.

In the learning phase, all training files are first used for the initialization of HMM models for each letter, using HTK HCompV. For each training word image, the corresponding sub-models are connected to form a right-left HMM. An embedded training using the Baum–Welch iterative estimation procedure is used with the HTK tool HERest. Using a training set, all the observation sequences are used to estimate the emission probability functions of each sub-model. The training procedure actually involves two steps that are iteratively applied to increase the number of Gaussian mixtures to a given M value. In the first step, a binary split procedure, along the iteration process, is applied to the Gaussians to increase their number. In the second step, the Baum–Welch re-estimation procedure is launched to estimate the parameters of the Gaussians. However, the expectation maximization (EM) algorithm is used to iteratively refine the component weights, means and variances to monotonically increase the likelihood of the training feature vectors

At recognition time, an ergodic HMM is formed using all sub-models. The recognition is done by selecting the best state sequence in the HMM using a Viterbi procedure implemented with the HTK tool HVite. Performances are evaluated in terms of word recognition rates using an unseen set of word images. The evaluation is obtained using the HTK tool HResult.

18.5.6 Experimental Results

Table 18.10 presents the DIVA-REGIM system results for the first APTI protocol (APTIPC1) for competition. The results are good for the majority of font sizes with an average of 98.95 % for word recognition rate and 99.88 % for character recognition rate. These results are better than those of the other participating systems in the ICDAR'2011 competition.

Table 18.11 presents the DIVA-REGIM system results for the second APTI protocol (APTIPC2) for competition. In term of results, DIVA-REGIM seems to be the best system compared to the other participating systems in the ICDAR'2011 competition with an average of 91.92 % and 97.72 % respectively for word recognition rate and character recognition rate.

Table 18.11 APTIPC2—DIVA-REGIM system results

Font		Size						Mean RR
		6	8	10	12	18	24	
Andalus	WRR	94.34	97.61	97.58	99.27	98.50	99.47	97.80
	CRR	97.94	99.26	99.45	99.70	99.49	99.82	99.28
Arabic Transparent	WRR	86.52	95.67	96.65	96.45	97.49	97.78	95.09
	CRR	93.87	98.51	99.13	99.10	99.40	99.25	98.21
Simplified Arabic	WRR	83.29	92.73	96.82	96.43	96.50	96.97	93.79
	CRR	92.16	97.37	98.99	98.82	99.13	98.71	97.53
Traditional Arabic	WRR	77.56	92.72	94.56	95.44	94.55	95.11	91.66
	CRR	96.03	98.87	98.92	98.94	99.00	98.79	98.43
Diwani Letter	WRR	57.47	80.50	84.18	89.88	90.08	85.46	81.26
	CRR	89.33	95.06	96.04	97.16	96.99	96.25	95.14
Mean RR of the system							WRR	91.92
							CRR	97.72

18.6 Conclusion

APTI is challenging, especially when we consider the recognition rate at the word level. APTI aims at a large-scale benchmarking of open-vocabulary text recognition systems. While it can be used for the evaluation of any OCR system, APTI is naturally well suited for the evaluation of screen-based OCR systems. The challenges addressed by the database are the variability of the sizes, fonts and styles, and the protocols that are defined are efficient enough to evidence the impact of such variability. The objective of the first competition, organized at the 11th International Conference on Document Analysis and Recognition (ICDAR'2011), in September 18–21, 2011, Beijing, China, for the recognition of multi-font and multi-size Arabic text, was to evaluate and compare different systems and approaches. We have presented in this chapter the results of four different systems on the ICDAR'2011 competition protocols with benchmarking strategy for Arabic low resolution word recognition.

Acknowledgements The authors would like to thank all ICDAR'2011—Arabic Recognition Competition: Multi-Font Multi-Size Digitally Represented Text participants and the anonymous reviewers for their comments and suggestions, which have much improved the presentation of this work.

References

1. Abbès, R., Dichy, J., Hassoun, M.: The architecture of a standard Arabic lexical database: some figures, ratios and categories from the DIINAR.1 source program. In: Proceedings of

the Workshop on Computational Approaches to Arabic Script-Based Languages, Semitic'04, pp. 15–22. Association for Computational Linguistics, Stroudsburg (2004)
2. Abdelraouf, A., Higgins, C.A., Khalil, M.: A database for Arabic printed character recognition. In: Proceedings of the 5th International Conference on Image Analysis and Recognition, ICIAR'08, pp. 567–578. Springer, Berlin (2008)
3. AbdelRaouf, A., Higgins, C., Pridmore, T., Khalil, M.: Building a multi-modal Arabic corpus (MMAC). Int. J. Doc. Anal. Recognit. **13**, 285–302 (2010)
4. Al-Muhtaseb, H.A., Mahmoud, S.A., Qahwaji, R.S.: Recognition of off-line printed Arabic text using hidden Markov models. Signal Process. **88**, 2902–2912 (2008)
5. Al-Sughaiyer, I.A., Al-Kharashi, I.A.: Arabic morphological analysis techniques: a comprehensive survey. J. Am. Soc. Inf. Sci. Technol. **55**, 189–213 (2004)
6. Baird, H.: The state of the art of document image degradation modelling. In: Chaudhuri, B.B. (ed.) Digital Document Processing, Advances in Pattern Recognition, pp. 261–279. Springer, London (2007)
7. Ben Hamadou, A.: A compression technique for Arabic dictionaries: the affix analysis. In: COLING'86, pp. 286–288 (1986)
8. Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum likelihood from incomplete data via the EM algorithm. J. R. Stat. Soc. B **39**(1), 1–38 (1977)
9. Dichy, J., Hassoun, M.: The DIINAR.1—Arabic lexical resource, an outline of contents and methodology. ELRA Newsl. **10**(2), 5–10 (2005)
10. Gimenez, A., Juan, A.: Embedded Bernoulli mixture HMMs for handwritten word recognition. In: Proc. of the 10th Int. Conf. on Doc. Analysis and Recognition (ICDAR), pp. 896–900 (2009)
11. Gimenez, A., Khoury, I., Juan, A.: Windowed Bernoulli mixture HMMs for Arabic handwritten word recognition. In: 2010 International Conference on Frontiers in Handwriting Recognition (ICFHR), pp. 533–538 (2010)
12. Graff, D., Chen, K., Kong, J., Maeda, K.: Arabic Gigaword, 2nd edn. Linguistic Data Consortium, Philadelphia (2006)
13. Hilal, Y.: Tahlil sarfi lil arabia. In: Proc. Comput. Process. Arabic Language, Kuwait (1985)
14. Jain, A.K., Duin, R.P.W., Mao, J.: Statistical pattern recognition: a review. IEEE Trans. Pattern Anal. Mach. Intell. **22**, 4–37 (2000)
15. Kanoun, S., Slimane, F., Guesmi, H., Ingold, R., Alimi, A.M., Hennebert, J.: Affixal approach versus analytical approach for off-line Arabic decomposable vocabulary recognition. In: ICDAR, pp. 661–665 (2009)
16. Kanoun, S., Alimi, A.M., Lecourtier, Y.: Natural language morphology integration in off-line Arabic optical text recognition. IEEE Trans. Syst. Man Cybern., Part B, Cybern. **41**(2), 579–590 (2011)
17. Khorsheed, M.S.: Offline recognition of omnifont Arabic text using the HMM toolkit (HTK). Pattern Recognit. Lett. **28**, 1563–1571 (2007)
18. Lee, C.H., Kanungo, T.: The architecture of TRUEVIZ: a groundtruth/metadata editing and visualizing toolkit. Pattern Recognit. **36**(3), 811–825 (2003)
19. Märgner, V., El Abed, H.: ICDAR 2009—Arabic handwriting recognition competition. In: ICDAR, pp. 1383–1387 (2009)
20. Märgner, V., El Abed, H.: ICFHR 2010—Arabic handwriting recognition competition. In: ICFHR, pp. 709–714 (2010)
21. Märgner, V., El Abed, H.: ICDAR 2011—Arabic handwriting recognition competition. In: 2011 International Conference on Document Analysis and Recognition (ICDAR), pp. 1444–1448 (2011)
22. Pechwitz, M., Maddouri, S.S., Märgner, V., Ellouze, N., Amiri, H.: IFN/ENIT—database of handwritten Arabic words. In: Proc. of CIFED 2002, pp. 129–136 (2002)
23. Rabiner, L., Juang, B.-H.: Fundamentals of Speech Recognition. Prentice Hall, Upper Saddle River (1993)
24. Schlosser, S.: ERIM Arabic database. Document Processing Research Program, Information and Materials Applications Laboratory, Environmental Research Institute of Michigan (1995)

25. Shaaban, Z.: A new recognition scheme for machine-printed Arabic texts based on neural networks. In: Proceedings of World Academy of Science, Engineering and Technology, vol. 31, July 2008
26. Shafait, F., Rashid, S.F., Breuel, T.M.: An evaluation of HMM-based techniques for the recognition of screen rendered text. In: International Conference on Document Analysis and Recognition, September 2011, pp. 1260–1264 (2011)
27. Slimane, F., Ingold, R., Alimi, A.M., Hennebert, J.: Duration models for Arabic text recognition using hidden Markov models. In: CIMCA, pp. 838–843 (2008)
28. Slimane, F., Ingold, R., Kanoun, S., Alimi, A.M., Hennebert, J.: Database and evaluation protocols for Arabic printed text recognition. In: DIUF—University of Fribourg, Switzerland (2009)
29. Slimane, F., Ingold, R., Kanoun, S., Alimi, A.M., Hennebert, J.: Impact of character models choice on Arabic text recognition performance. In: International Conference on Frontiers in Handwriting Recognition (ICFHR), Novemberf 2010, pp. 670–675 (2010)
30. Slimane, F., Kanoun, S., Alimi, A.M., Ingold, R., Hennebert, J.: Gaussian mixture models for Arabic font recognition. In: ICPR, pp. 2174–2177 (2010)
31. Wachenfeld, S., Klein, H.-U., Jiang, X.: Recognition of screen-rendered text. In: Proceedings of the 18th International Conference on Pattern Recognition—Vol. 02, ICPR'06, pp. 1086–1089. IEEE Comput. Soc., Washington (2006)
32. Young, S.J., Evermann, G., Gales, M.J.F., Hain, T., Kershaw, D., Moore, G., Odell, J., Ollason, D., Povey, D., Valtchev, V., Woodland, P.C.: The HTK Book, Version 3.4. Cambridge University Engineering Department, Cambridge (2006)