Published version: http://www.sciencedirect.com/science/article/pii/S0031320315003428

Open-Vocabulary Recognition of Machine-Printed Arabic Text Using Hidden Markov Models

Irfan Ahmad [1,2,*], Sabri A. Mahmoud [1], and Gernot A. Fink [2]

[1]Information and Computer Science Department, KFUPM, Dhahran Saudi Arabia
[2] Department of Computer Science, TU Dortmund University, Germany

Abstract—In this paper, we present multi-font printed Arabic text recognition using hidden Markov models (HMMs). We propose a novel approach to the sliding window technique for feature extraction. The size and position of the cells of the sliding window adapt to the writing line of Arabic text and ink-pixel distributions. We employ a two-step approach for mixed-font text recognition, in which the input text line image is associated with the closest known font in the first step, using simple and effective features for font identification. The text line is subsequently recognized by the recognizer that was trained for the particular font in the next step. This approach proves to be more effective than text recognition, which employs a recognizer trained on samples from multiple fonts. We also present a framework for the recognition of unseen fonts, which employs font association and HMM adaptation techniques. Experiments were conducted using two separate databases of printed Arabic text to demonstrate the effectiveness of the presented techniques. The presented techniques can be easily adapted to other scripts, such as Roman script.

Keywords—*Optical character recognition, mixed-font OCR, unseen-font OCR, hidden Markov models, font identification, sliding window, Arabic OCR.*

1. Introduction

In this digital age, seamless interaction between the physical world and the digital world is a primary objective. Digitizing documents into an electronic form that can be easily stored, retrieved, searched and indexed is critical. Due to the widespread use of paper and because vast amounts of information are already available in paper form, the need to convert this information into electronic form [1] has prompted the need for highly reliable and robust document analysis and processing systems. The core component of a document processing system is the text recognition module. The success of any document processing system requires a highly accurate text recognition module. Separate systems are generally trained and employed for handwritten versus printed text recognition tasks.

[*] Corresponding author, P.O. Box. 303, KFUPM, Dhahran, Saudi Arabia, 31261. Phone: +966-13-8601243, Fax: +966-13-8602174.

Email addresses: irfanics@kfupm.edu.sa (Irfan Ahmad), smasaad@kfupm.edu.sa (Sabri A. Mahmoud), gernot.fink@tu-dortmund.de (Gernot A. Fink)

Although printed text recognition is more developed than handwritten text recognition, it presents challenges that need to be addressed. The main challenges are related to the recognition of text in degraded documents, irregular and unaligned text, and mixed fonts. If the text to be recognized has a font typeface (referred to as font) that substantially differs from the fonts on which the recognizer was trained, the complexity of the task increases.

Hidden Markov models (HMMs) are one of the most extensively used and successful classifiers for text recognition [2], [3]. They prevent the need to explicitly segment text line images into smaller units, such as characters or strokes, which is common when using other classifiers. HMMs can seamlessly integrate and apply language models during the decoding process. In addition, it has sound theoretical and mathematical foundations and can robustly manage noise. The general trend for Arabic text recognition is to use HMMs due to the cursive nature of Arabic text (in addition to the reasons that were previously cited). Although printed Arabic text recognition encounters challenges that are similar to challenges encountered by other scripts, it has a unique set of peculiarities. These peculiarities include the right-to-left writing direction, which can be easily adapted using an existing recognizer that was designed for other scripts, such as Roman script [4][5]. Other distinctive features of Arabic script create possibilities for researchers to investigate how to address them and whether they can be utilized as leverage.

Arabic script is cursive in both printed and handwritten form. It has 28 basic characters; of these, 16 characters have one or more dots either above or below them. These dots differentiate the similar core shapes. Some characters can connect to the subsequent characters in a word, whereas other characters can only be connected to but they cannot connect to subsequent characters in a word. The shape of an Arabic character is dependent on its position in the word. Some characters (characters that can connect to the subsequent characters in a word) can assume a maximum number of four position-dependent shapes, whereas other characters (characters that cannot connect to subsequent characters) have two position-dependent shapes. Optional diacritics exist that can be attached either above or below the characters. These diacritics differ from the mandatory dots that separate different characters with similar core shapes. Another important aspect of the script is its prominent writing line; Arabic script has a sharp writing line. Figure 1 displays sample text in Arabic and Roman scripts and their projections. The Arabic script has a prominent and sharp writing line that has a unique pattern of pixel distribution. These properties of the script can be utilized for robust and adaptive cell division of the sliding windows that are employed for feature extraction.

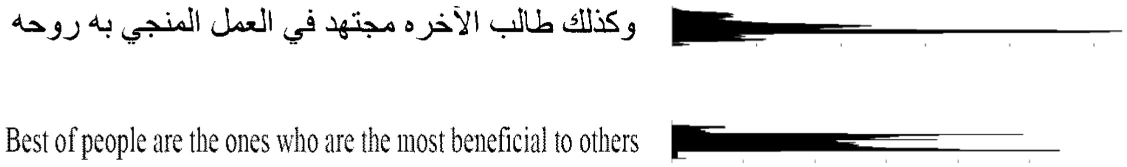


Figure 1: Projection profiles for printed text in Arabic script (top) and Roman (bottom) script.

In this paper, we present printed Arabic text recognition using HMM. Our text recognition task is performed using continuous text lines images in multiple fonts instead of isolated digit, character, or word images. The text comprises open vocabulary text recognition in which no word lexicon is utilized during the recognition. We also evaluated our system using the publicly available Arabic Printed Text Image (APTI) database of printed Arabic words [6]. Experiments were performed using this publicly available database to demonstrate the robustness of the presented techniques.

We present a new approach to the sliding window technique for feature extraction. The size and position of the cells in the sliding window adapt to the text line image depending on the writing line of the Arabic text and the ink-pixel distributions. We also propose some simple and effective features for font identification, which are primarily designed based on the projection profiles of Arabic script. The font identification step integrates with our printed text recognition framework for mixed-font text recognition and for unseen-font text recognition tasks. We employ a two-step approach in which the input text line image is associated with the closest known font in the first step and HMM-based text recognition is performed in the second step using the recognizer that was trained on the associated text font. This approach proved to be more effective than the commonly employed approach for recognizing text using a recognizer that was trained on text samples from multiple fonts [7], [8]. Our approach overcomes the common limitations of other techniques, such as the need for labeled samples of the text images in the font to be recognized and the assumption of data isogeny, i.e., the text lines to be recognized are obtained from only one font at a time [9].

This paper is organized as follows: The next section presents related studies on printed text recognition, which primarily focus on printed Arabic text recognition. In Section 3, we present our approaches and techniques for printed Arabic text recognition using HMMs. We present our adaptive sliding window technique, the two-step approach to font-association-based text recognition, and a framework for unseen-font text recognition. In Section 4, we present the experiments and the results. In Section 5, we present a comparison of our work with similar works on printed Arabic text recognition. In Section 6, we present the conclusions of our work.

2. Related Work

Research in optical character recognition began in the 1940s, and commercial optical character recognition (OCR machines) appeared in the 1950s [10]. Previous systems were substantially restricted in terms of operating conditions, document layout, and the fonts that can be recognized. Current systems enable flexible operating conditions and the ability to address complex document layouts and varied fonts (e.g., [11], [12]). One of the earliest studies on Arabic OCR was conducted in the 1970s (cf. [1]). Interest in research on Arabic text recognition and related applications has considerably increased in the last decade, as indicated by the number of publications that have resulted from this research. In this section, we primarily limit our discussion to related studies that employ HMMs for two reasons: (a) because HMMs are popular text recognition techniques and (b) because Arabic script is cursive and HMMs are primarily employed for Arabic text recognition to prevent the need to explicitly segment images beyond text lines. For a broader perspective on text recognition, readers can refer to [1], [10], [13]–[19].

Bazzi et al. [4] investigated omni-font text recognition for English and Arabic; a text recognition system was adapted from their HMM-based speech recognition system. *Bakis* topology was employed with the same number of states for all models. Each Arabic character shape was modeled with a separate HMM. The authors introduced six additional models for six common ligatures that appeared in printed Arabic text. They proposed the careful distribution of training data based on different styles (e.g., bold and italics) to prevent the recognizer from bias toward the dominant style of the training data. The results for mixed-font recognition were deficient compared with the average results for mono-font recognition, which is understandable. No special treatment for mixed-font text recognition was proposed, with the exception of training the recognizer on text images from multiple fonts to enable the model to sufficiently generalize. Khorsheed presented a discrete HMM-based system for printed Arabic text recognition [20]. The sliding window was vertically divided into a specific number of cells. Pixel density features were calculated from each sliding window cell and concatenated as feature vectors; these features were later discretized. The majority of the system's characteristics are similar to the characteristics of the system that was presented in [4], with the exception that the system was based on discrete HMMs. Experiments were conducted on a database of six different fonts; no special treatment was proposed for mixed-font text recognition.

Natarajan et al. [21] presented an HMM-based OCR system for multiple scripts. The majority of the system components were adapted from their speech recognition system, with the exception of feature extraction. They presented pixel percentile features as a novelty. These features are robust to image noise. Pixels are accumulated from the top of a sliding window frame to the bottom of a sliding window frame. The image height at a certain pixel percentile is considered to be a feature. The values at 20 equally separated pixel percentiles (from 0 to 100) are appended to form a feature vector, and horizontal and vertical derivatives of these features are also appended to the feature vector. In addition, they compute angle and correlation features from ten cells of a window frame (window frames are divided into ten overlapping cells from top to bottom). They demonstrated the effectiveness of their features and the overall OCR system by recognizing the text from three different scripts: English, Arabic, and Chinese. Unsupervised HMM adaptation was employed for the text recognition of documents with fax-related degradation.

Prasad et al. [8] presented some improvements to the Arabic OCR system of the BBN Technologies. They presented the use of the parts of Arabic words (PAW) language models, which demonstrated better performance in terms of recognition rates, which exceeded the performance of word and character language models. Position-dependent HMMs, in which each Arabic character shape is treated as a separate HMM, were compared with position-independent models, in which each Arabic character had only one model. In addition, contextual tri-character HMMs were also investigated. The use of position-dependent HMMs yielded better results than the results of position-independent HMMs. However, contextual modeling and position-dependent HMMs reduced the recognition rates. Contextual HMMs for the position-independent approach improve the results compared with the results from simple position-independent modeling, which is understandable. Thus, the use of position-independent HMMs may be sufficient for capturing the contextual variations in printed Arabic text recognition. The study did not reveal any special strategies for addressing text recognition in multiple fonts.

Al-Muhtaseb et al. proposed a hierarchical sliding window for printed Arabic text recognition [22]. A window is divided into eight non-overlapping vertical segments. Eight features (including ink pixels) are extracted from these eight segments. Four additional features are computed from the eight features using virtual vertical sliding windows, in which the height of a window is one-fourth the height of the writing line. Three additional features are calculated using a virtual vertical overlapping sliding window height that is one-half the writing line height with an overlap of one-fourth the writing line height. An additional feature is computed by summing the first eight features. These hierarchical windows generate features with greater weight in the center region of the writing line (baseline). These hierarchical windows produced very high recognition rates with synthesized data [22]. Experiments with text line images extracted from scanned documents yielded poor results [23].

Slimane et al. [24] proposed certain font-specific features for complex Arabic fonts, such as DecoType Thuluth, DecoType Naskh, and Diwani Letters. These fonts are complex due to their complex appearances and ligatures. The authors proposed a large number of features; some of the features are common to all fonts, whereas other features are specific to each font. HMMs were employed as the recognition engine, and the authors obtained reasonable improvements over the baseline system for the three fonts. The authors evaluated their system using an APTI database of low-resolution printed

Arabic words in multiple fonts with different degradation conditions [6]. Because the database was synthetically generated, how the presented system will perform on a real scanned database is anticipated.

Ait-Mohand et al. [9] recently presented an interesting study of mixed-font text recognition using HMMs. The main contribution of the study was related to HMM model length adaptation techniques that were integrated with HMM data adaptation techniques, such as maximum likelihood linear regression (MLLR) and maximum a posteriori (MAP) techniques. The proposed techniques were effective in mixed-font text recognition tasks, and the authors reported significant improvements with their technique compared with traditional HMM adaptation, which only addresses the data aspect of HMM. The authors noted two main limitations of the study: the need for small amounts of labeled data for the evaluation font and the assumption that all line images during the evaluation would be obtained from a single font.

3. Printed Arabic Text Recognition

In this section, we present our contributions to printed Arabic text recognition. We present a brief description of HMM-based printed Arabic text recognition and present our new adaptive sliding window technique for feature extraction. In Section 3.2, we describe mixed-font text recognition tasks. In Section 3.3, we present text recognition of unseen font text and present an overall framework for the task.

3.1. HMM-Based Printed Arabic Text Recognition

In this section, we present details on HMM-based text recognition that are applied to the recognition of printed Arabic text line images. Note that our text recognition task pertained to continuous text line images, which included running text instead of isolated digit, character, or word images. This text comprised open vocabulary text recognition in which no word lexicon was employed during the recognition.

Training a typical HMM-based text recognizer involves three main steps: preprocessing, feature extraction, and training the HMMs. The first step involves preprocessing, which primarily involves preparing the text line images for feature extraction. Scanned page images may undergo a number of optional processing steps, including binarization, skew correction, and noise removal, and the text line images are extracted from the page image. Alternatively, many of these processing steps can be directly performed on the text line images instead of applied at the page level. Text line images are optionally normalized to a certain height before the features are extracted.

In feature extraction for HMM-based text recognition, a sliding window with the height of the line image is run across the image from one end to the other end (in the writing direction) and a number of features are computed from the image portion under the window. The width of the window and the optional overlap are empirically established.

The window may be divided into a number of vertical cells. In the next section, we present our novel adaptive sliding window for feature extraction, which we believe is more suitable for Arabic script. Different types of features have been employed for text recognition, such as pixel density, geometric features, and Gabor-filter-based features.

In the training step, the transcription from the training text images and other features are employed to train the HMMs that represent the basic recognition units, such as characters. The selection of a basic recognition unit is interesting in the case of Arabic because Arabic text is cursive in both printed and handwritten forms. The majority of the characters in Arabic can assume four different visual shapes depending on their position in the text, i.e., *alone*, *beginning*, *middle*, and e*nding*. Some Arabic characters only assume *alone* and *ending* shapes because no characters can directly connect after them. Due to this variability in character shapes, the most common approach is to model each position-dependent character shape as a separate HMM and subsequently map it to the corresponding character after recognition. Although this approach is suitable in terms of recognition performance, it causes a fourfold increase in the number of HMMs that need to be trained compared with the selection of characters as the basic HMM units. Other approaches for modeling basic HMM units for Arabic script have also been presented in the literature [25]–[27]. In this study, we treat each character shape as an HMM unit. In the case of text line recognition, modeling space is also important. In this study, we employ a space model along with character-shape HMMs.

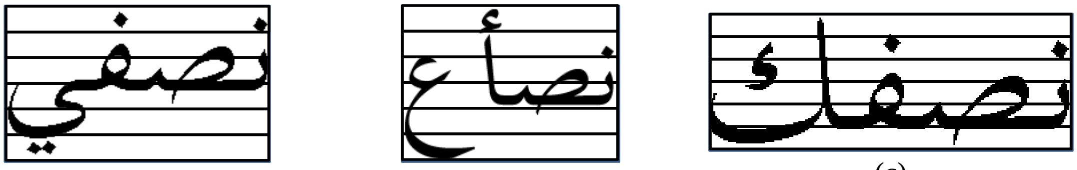
Many training algorithms are available; the most commonly applied model is the Baum-Welch algorithm, which is based on expectation maximization (EM). As an extension to the basic uniform initialization approach, a two-step approach is also commonly employed by which the recognizer that was trained using a uniform initialization approach is employed to annotate the character boundaries in the text image. This information is utilized to retrain the recognizer with Viterbi initialized models. In this study, we employ the two-step training approach.

Recognition using HMM-based systems involves three main steps: preprocessing, feature extraction, and decoding (i.e., generating hypotheses from text images). The first two steps are similar to the training stage. The trained HMMs generate the recognition hypothesis using the features extracted from the images of the evaluation set. Viterbi decoding is the most commonly used algorithm. Optionally, language models can be employed while decoding. Statistical n-gram language models are the most common language models. These models are generally estimated from the training-set transcriptions and certain external, large-text corpora. Statistical n-grams can also be employed after the decoding stage to re-score the optional N-best lists that were generated during decoding. For an in-depth discussion on the use of HMMs for text recognition and related fields, interested readers can refer to [2], [3], [28].

3.1.1. Adaptive Sliding Window for Printed Arabic Text Recognition

The use of a sliding window for feature extraction from the text line images is the most common approach when using HMMs for recognition [2]. It enables the sequencing of two-dimensional image data and prevents the need to segment text lines into characters or smaller units. Different approaches exist for designing sliding windows for extracting features from text line images. One of the earliest approaches for printed text recognition was presented by the BBN group [4]. A window frame ran along the text line image. The width of the frame consisted of a small number of pixels; part of it overlapped with the next frame. Each frame was vertically divided into multiple cells of uniform size. Simple features were extracted from each of these cells and concatenated to form the feature vector for a sliding window frame. This effective design has been used employed by researchers for printed text recognition [20], [29], [30]. Muhtaseb et al. [22] proposed a hierarchal window scheme for printed text recognition that exhibited some similarities with the previous approach but also differed with the approach because it not only extracted features from the individual cells but also successively combined different cell groups in a frame and extracted additional features from them. Other approaches exist in which the sliding window is not subdivided into cells but the features are extracted from the complete window [29]. Alhajj et al. [31] proposed the use of slanting windows in addition to typical straight windows for handwritten text recognition to capture writing variability caused by slants in handwriting and the overlap and shifting positions of diacritical marks.

Dividing the sliding window into a number of cells, as discussed in [4], [20], [22], is not the best option for Arabic script, as discussed in [23]. The vertical positions of characters in a text line image may vary depending on the actual text and the presence or absence of optional diacritics. Figure 2 illustrates this issue with some examples of Arabic word images. The same character can be positioned in different cells in different words. The dot for the first character (from the right) *Noon (ن)* as shown in Figure 2 (a), is located in cell one but the core is primarily located in cell three. The same character in Figure 2 (b) has its core in cell four with the dot in cell two. In Figure 2 (c), the character *Noon* has its dot in cell two but the majority of its core is in cell five. Similar observations of other characters can be performed.



(a) (b) (c)

Figure 2: Illustrations that explain the issues of uniform cell division for Arabic script.

We propose an improvement to the sliding window cell division technique. We apply the peculiarities of Arabic script to design a new technique for cell division. Arabic script has a very prominent writing line. We observe a sharp increase in pixel density and a sharp decrease toward the lower half of the line. This property of the script was utilized to determine cell positions that are robust to variations in the writing line position with respect to the image's height. The cell sizes vary such that the cells are smaller around the writing line where the pixel concentration is higher and gradually increase as we move away from the writing line (both below and above it). A cell is placed around the writing line and a number of cells are placed above and below it. The number of cells below the writing line is usually less than the number of cells above the line because this design is suitable for the properties of Arabic script.

Figure 3 presents the algorithm for cell division using the sliding window. The user selects the total number of cells and the number of cells above the writing line. Once the cell division for the sliding window has been established, the desired features can be extracted.

```
User Inputs:= total no. of cells 'totalCells' in a frame,
              no. of cells 'cellsAbove' above the writing line,
              text-line image

•  Place the cell $cell_{wl}$ around the writing line such that the baseline of
   the text is in the middle of the cell. The width of the cell shall be
   such that the following condition is satisfied:

   $\frac{(Sum\ of\ ink\ pixels\ within\ the\ cell)}{(total\ ink\ pixels)} \cong \frac{1}{(total\ no.of\ cells\ in\ a\ frame)}$

•  Divide the area above the $cell_{wl}$ into 'cellsAbove' cells such that each
   'cellsAbove' cell has the same percentage of ink pixels.

•  Divide the area below the $cell_{wl}$ into 'totalCells – cellsAbove - 1' cells
   such that each 'totalCells – cellsAbove -1' cell has the same percentage
   of ink pixels.
```

Figure 3: Algorithm for determining the size and position of cells on a text line image.

As discussed in the experimental results section, this technique is effective for printed Arabic text recognition. We can achieve fairly satisfactory results using the presented sliding window technique when used even with simple features like pixel densities.

3.2. Mixed-font Text Recognition

In practical situations, the expectation of recognizing text from only one font may be simplistic. Conversely, the recognition of text in multiple fonts is possible and the font order of the text line images may be random. To contribute to these conditions, recognition of the text in fonts that were not observed during training is expected. We address mixed-font text recognition in this section, and text recognition for unseen fonts will be addressed in the next section.

Researchers have addressed mixed-font text recognition using various approaches. The most common approach is to train the recognizer with samples from as many fonts as possible to possibly address the variability during the recognition phase [4], [8], [20], [32]. This approach may result in better overall recognition compared with recognizing mixed-font text using a recognizer that was trained in only one font. However, the error rates are significantly higher than the mean error rates for mono-font text recognition.

Another approach for addressing mixed fonts was proposed by Ait-Mohand et al. [9]. They proposed HMM adaptation techniques in which the adaptation was performed using the HMM data and the model length (number of states). They demonstrated the effectiveness of their technique for mixed-font and unseen-font text recognition. However, the technique exhibited two major issues: the need for a few labeled text samples in the recognition font, i.e., the need for adaptation data, and the dependency of the technique on the assumption that all text line images to be recognized will belong to a single font. To overcome the limitations and effectiveness of the previous techniques, we employ a two-step font association approach, as discussed in the subsequent section.

3.2.1. Font Association Based Text Recognition

To address the situation of mixed-font text recognition, we employ two-step font association recognition. We propose training mono-font text recognizers instead of training a recognizer on text images from multiple fonts. We propose a font identification module that can associate a text line image to the closest trained font. During recognition, the input text line image will be associated with a font. As a second step, we employ the mono-font recognizer, which is trained on the associated font, to generate the recognition hypothesis. As demonstrated in Section 4.1.3, this approach is very effective and the error rates in mixed-font scenario can approach close to the error rates achieved in mono-font text recognition tasks. In addition, this approach enables the use of font-specific parameters (if any) for feature extraction and training, which can optimize the recognition performance. To train the font association module, appropriate features and classifiers can be utilized. In this study, we present a set of simple and effective features for font identification that are primarily dependent on the projection profile of the text line image. These features were employed with support vector machine (SVM) classifiers, and the font identification results are very promising, as demonstrated in Section 4.1.3. We describe our font identification features in the next section.

Font Identification Features

We propose the following features for identifying the font in a text line image. The features are extracted from height-normalized (maintaining a constant aspect ratio) text line images. Prior to introducing the features, we introduce the function *p(i, j)*, which we frequently employ to define our features:

$$p(i,j) = \begin{cases} 1, & if\ row\ 'i'\ column\ 'j'\ of\ the\ image\ has\ ink\ pixel \\ 0, & otherwise \end{cases}$$

We employ the term *'h'* to denote image height and '*w'* to denote image width.

a. ***Maximum ink projection (F₁)***: This feature calculates the maximum value of the ink projection of the text image. The value is normalized by the image width. The dimension of the feature is one.

$$F_1 = \frac{\max\limits_{i:=1\ to\ h} \left(\sum_{j=1}^{w} p(i,j)\right)}{w}$$

b. ***Ratio of ink pixels (F₂):*** The ratio of the number of ink pixels in a row to the maximum ink projection. The dimension of the feature is identical to the normalized height of the image.

$$F_2(i) = \frac{\sum_{j=1}^{w} p(i,j)}{\max\limits_{i:=1\ to\ h} \left(\sum_{j=1}^{w} p(i,j)\right)}$$

c. ***Percentage increase/decrease of pixel projection (F₃)*****:** The percent increase or decrease in the pixel projection in a given row compared with the row immediately above it. The dimension of the features is one less than the normalized height of the image.

$$F_3(i) = \frac{\sum_{j=1}^{w} p(i,j) - \sum_{j=1}^{w} p(i-1,j)}{\sum_{j=1}^{w} p(i-1,j)} \qquad ; where\ 1 < i\ \leq h$$

d. ***Compaction (F₄):*** The ratio of the total number of ink pixels in a text line image to the total area of the line image. The dimension of the feature is one.

$$F_4 = \frac{\sum_{i=1}^{h} \sum_{j=1}^{w} p(i,j)}{h \times w}$$

e. ***Count of projections above average (F₅):*** The count of the number of rows in the image in which the ink-pixel count exceeds the average ink-pixel count of the image rows. The dimension of the feature is one.

$$F_5 = \sum_{i=1}^{h} a(i)\,; \quad where,$$

$$a(i) = \begin{cases} 1, & if \displaystyle\sum_{j=1}^{w} p(i,j) > \frac{\sum_{i=1}^{h} \sum_{j=1}^{w} p(i,j)}{h} \\ \\ 0, & otherwise \end{cases}$$

We concatenate these defined features into one feature vector for a text line image.

3.3. Text Recognition of Unseen Font

As mentioned in the previous section, we should recognize input text with previously unseen fonts, i.e., when we have no training samples for a specific font. This issue is one of the most difficult problems to address. A simple approach to the problem is to use a recognizer that was trained on as many font samples as possible to recognize the unseen-font text. Another improvement over the previous approach is to use HMM adaptation techniques for the unseen font. If we can arrange for a small number of labeled samples of the unseen font, supervised adaptation (such as MLLR) would be an acceptable option, whereas unsupervised adaptation techniques can be employed if no labeled samples were available. We propose a two-step approach, in which we associate the input text line image with the closest known font and adapt the font-specific recognizer using HMM adaptation techniques.

3.3.1. HMM Adaptation

HMM adaptation techniques have been successfully employed in speech recognition [33], [34]. Instead of training the recognizer for a particular speaker, which can require a substantial amount of data and may not be feasible, a small amount of speaker-specific data can be utilized to adapt the model parameters of a general recognizer to fit the speaker-specific characteristics. If the speaker-specific labeled data are available prior to recognition, then *supervised* adaptation can be performed. If no data are available pre-recognition, then *unsupervised* adaptation can be performed during the recognition process using the recognition output as the labeled adaptation data to be applied in subsequent recognition [33]. The same idea of HMM adaptation has been successfully extended to the domain of text recognition. It has been employed for adapting handwritten text recognizers for new writers [30] and for improving recognition accuracy on fax-degraded documents [21]. HMM adaptation techniques have been applied to adapt a general text recognizer to a specific font [9]. In general, model parameters related to the data are adapted but the model length and transition probabilities are not modified. However, a recent study proposed methods for integrating model length adaptation with parameter adaptation [9]. In the remainder of the paper, we focus our discussion on parameter adaptation.

The task of adaptation is to obtain new model parameters $\hat{\theta}$ by fine-tuning the original model parameters θ to maximize the likelihood of adaptation data O.

$$\hat{\theta} = \arg\max_{\theta} p(\theta|O)$$

The model parameters that are generally adapted include the mixture means μ and variances Σ; in [21], it was only utilized to adapt means. MLLR is one of the most common techniques for HMM adaptation. MLLR estimates linear transformations for means and variances to adjust them to better fit the adaptation data. To robustly estimate the transformations given the limited availability of adaptation data, transformations are linked across multiple Gaussians. A group of Gaussians that share the same transform are referred to as a *regression class*. For details about HMM adaptations using MLLR, readers can refer to [33], [34].

To recognize text from unseen fonts, we propose a two-stage approach. First, we associate the input text image to be recognized with the closest font in our trained model set. Second, we adapt the associated font's recognizer for the unseen font using MLLR-based HMM adaptation techniques. We perform supervised or unsupervised adaptation depending on the availability of labeled samples for the unseen font. We discovered (as demonstrated in Section 4.1.4) that adapting the closest font recognizer is more effective than adapting a recognizer that was trained on multiple fonts.

3.3.2. The Framework

Based on these discussions, we propose a framework for printed text recognition using the HMM. In the first step, we train an HMM recognizer for individual fonts. We also train the font association module using training samples from individual fonts. Features such as the ones described in Section 3.2.1 are employed to train suitable classifiers (such as SVMs, random forests, and HMMs). For an input text line image that has to be recognized, we associate the text image with the closest known font. Next, we extract text recognition features from the text line image. If font-specific parameters for feature extraction (such as window width and overlap) exist, they can be employed during feature extraction. After feature extraction, we employ the associated font's HMM recognizer for decoding. If we expect a batch of text line images from a single font during decoding, we have the following two options: (i) If some labeled samples are available for the input text's font, we perform supervised HMM adaptation prior to decoding; otherwise, (ii) we perform unsupervised adaptation during decoding. If we expect random input images from a number of fonts, then we decode the text using the associated font's recognizer. Figure 4 illustrates the framework steps.

In the case in which the unseen font significantly differs from any of the trained fonts, the text line image may not be associated with only one font with high confidence. In these cases, a group of fonts (that is, a subset of all trained fonts) may be more representative of the text line font than a single font. A recognizer that has been trained on the subset of the fonts can be employed instead of a recognizer that was trained on a single font. Investigating this approach is a future work.

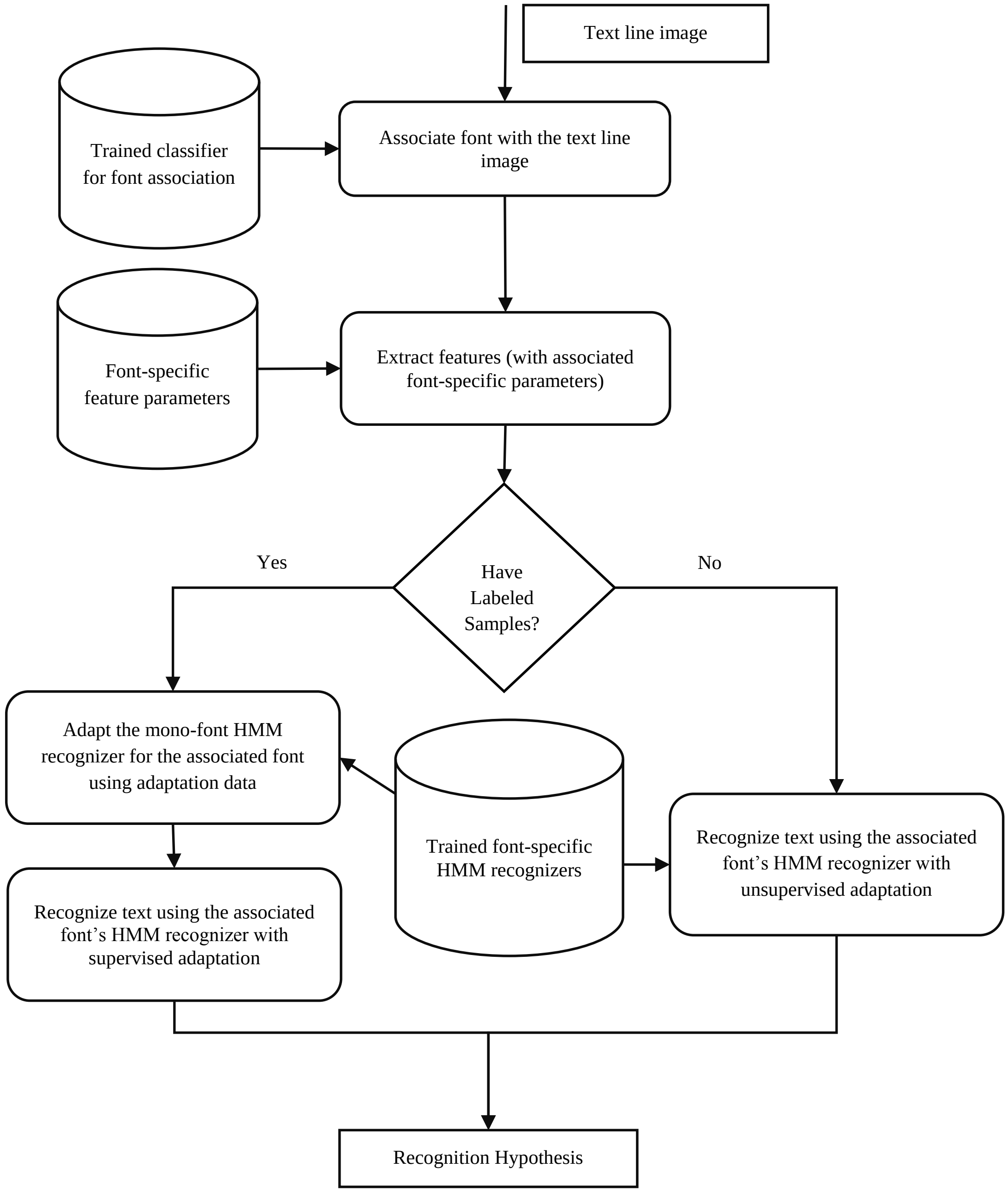

Text line image
Trained classifier for font association
Associate font with the text line image
Font-specific feature parameters
Extract features (with associated font-specific parameters)
Yes
Have Labeled Samples?
No
Adapt the mono-font HMM recognizer for the associated font using adaptation data
Trained font-specific HMM recognizers
Recognize text using the associated font's HMM recognizer with unsupervised adaptation
Recognize text using the associated font's HMM recognizer with supervised adaptation
Recognition Hypothesis

Figure 4: Framework for printed text recognition.

4. Experiments and Results

In this section, we describe the experiments that we performed and discuss the results. We performed experiments using two different databases of printed Arabic text. As there is no printed, text-line Arabic database that is publicly available for the research community, we developed our own database—the Printed-KHATT (P-KHATT) —for research on printed Arabic text recognition at the text-line level in multiple fonts. We present the database that we developed and the experiments that we conducted using this database. To obtain some comparative results for our OCR system and the presented techniques, we also conducted text recognition experiments using the publicly available APTI database of printed Arabic words [6]. The details on experiments conducted on the APTI database is presented in Section 4.2.

4.1. Text Recognition Using the P-KHATT Database

First, we describe the P-KHATT database that we developed to conduct experiments and investigate the performance and effectiveness of our technique. We include some important statistics about the database and the setups that were employed for experimentation. Second, the mono-font text recognition experiments, in which we employed one font at a time, and the mixed-font text recognition experiments, in which the task was to recognize text from text line images from multiple fonts, are discussed. Last, we detail the experiments that we conducted to recognize text from unseen fonts, i.e., no text line training images were available for the respective font.

4.1.1. The P-KHATT database of printed Arabic Text

The majority of the publicly available databases incorporate digits, isolated characters, and isolated words. We developed a multi-font printed Arabic text database—the P-KHATT database—for research in the area of printed text recognition. The P-KHATT database is based on the KHATT database of unconstrained handwritten Arabic text [35], [36]. The database includes text from eight different fonts; each text is divided into three non-overlapping sets (training, development, and evaluation). The text and the divisions are similar to the text and divisions of the KHATT database. Figure 5 presents sample text images from the P-KHATT database in eight fonts.

Text was printed and scanned at 300 dots per inch (DPI). Scanned pages were skew-corrected using the technique presented in [37], and the text line images were segmented from the skew-corrected page images. Table 1 presents some useful statistics from the P-KHATT database. In addition to the data and images for the eight fonts, the P-KHATT database has text line images and labels for a ninth font for the purpose of text recognition for unseen fonts. The ninth font does not include the training and the development sets.

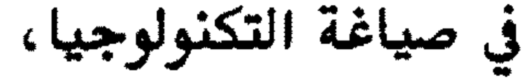
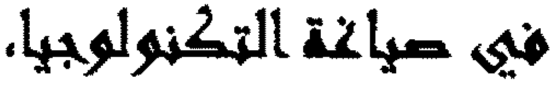
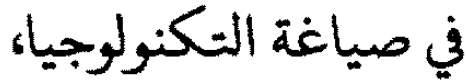
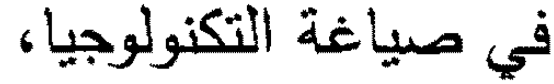
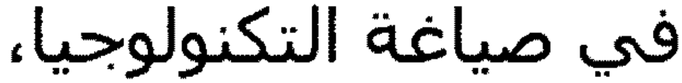
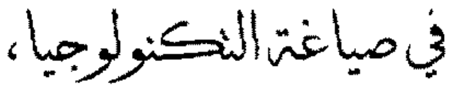
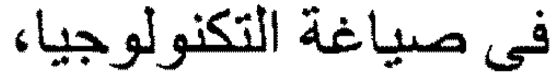
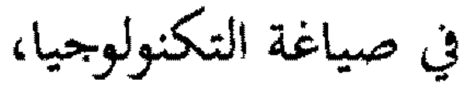
Font (Code)	*Sample Text Image*
Akhbar (AKH)	 في صياغة التكنولوجيا ،
Andalus (AND)	 في صياغة التكنولوجيا،
Naskh (NAS) *(KFGQPC Uthman Taha Naskh)*	 في صياغة التكنولوجيا،
Simplified Arabic (SIM)	 في صياغة التكنولوجيا ،
Tahoma (TAH)	 في صياغة التكنولوجيا،
Thuluth (TLT) *DecoType Thuluth*	 في صياغة التكنولوجيا ،
Times New Roman (TNR)	 في صياغة التكنولوجيا،
Traditional Arabic (TRA)	 في صياغة التكنولوجيا،

Figure 5: Sample text line images from the P-KHATT database in different fonts. Image degradation due to the printing and scanning process is distinct.

Proposed setups of the experiments

We propose three different setups for the experiments with the database. In the first setup, each of the eight different fonts has individual training, development, and evaluation sets. This setup is employed for mono-font recognition. A suitable text recognizer is expected to have reasonable recognition rates for each font when they are separately trained and evaluated. In the second setup, the development and evaluation sets have samples from all eight fonts. An equal number of samples from each font was selected and randomly sequenced in the development and evaluation sets. This setup enabled us to evaluate a mixed-font recognition system. The last setup had an unseen-font evaluation set, i.e., its font differed from the eight available fonts. This setup enabled us to evaluate the robustness of the recognizer in situations with limited or no samples from the font with which text line images needed to be recognized.

Table 1: Some useful statistics about the P-KHATT database (per font).

Set	*No. of lines*	*No. of words*	*No. of characters (without spaces)*	*No. of characters longest line*	*No. of characters shortest line*
Training	6,472	75,216	358,554	132	1
Development	1,414	16,019	77,558	126	4
Evaluation	1,424	15,710	76,571	135	3
All data	9,310	106,945	512,683	135	1

4.1.2. Mono-font Text Recognition

In this section, we present mono-font text recognition using the adaptive sliding window for feature extraction. We normalized the line images of the P-KHATT database to a fixed height (96 pixels) while maintaining a constant aspect ratio. Next, we extracted the features from the normalized text line images. We employed simple pixel density features from the text line image and its horizontal and vertical edge derivatives. We divided the sliding window into six cells; three cells above the writing line, one cell around the writing line, and the remaining two cells below the writing line. Thus, the dimension of the feature vector was eighteen (six from the image and six each from the horizontal and vertical derivatives of the images).

Our text recognition system is based on continuous-density hidden Markov models (HMMs). We use HTK tools [28] to implement our recognizer. Each character shape is treated as an individual model with a simple right-to-left linear topology. Thus, some characters (such as *Seen* س, *Jeem* ج) have four different models because they can assume four different position-dependent shapes. Other characters (such as *Waw* و, *Daal* د) have two models because they can only assume two position-dependent shapes. White space was explicitly modeled using a separate HMM. In addition, some non-Arabic characters and digits in the database, each of which has a separate HMM, produced a total of 153 different HMMs in our recognition system. Note that the different character shapes were merged to the corresponding characters after the recognition because they essentially represented the same character. This technique is more effective than treating each character as a class irrespective of its shape. Each character-shape HMM was modeled with the same number of states, with the exception of some narrow-width characters (such as *Alif* ا), which were modeled with half the number of states. The optimal number of states (for each font) was determined based on the uniform initialization (flat start) recognition results on the font’s development set.

We employed 2,000 text line images for training instead of the complete training set for each font. Training was conducted in two stages. In the first stage, uniform initialization (flat start) was performed using the training data. In the next stage, the alignment information from the training data was employed to initialize individual HMMs using Viterbi initialization followed by a number of iterations of Baum-Welch retraining. Character hypotheses for the evaluation set were generated using Viterbi decoding.

Although we could have optimally adjusted the values for the sliding window width and overlap for each font based on the recognition results from the development set for each font, we employed the same values for all fonts (with the exception of the Thuluth font, which will be subsequently discussed). The optimal values for the sliding window width and overlap were determined based on the recognition results from the development set for the Times New Roman font. Table 2 presents the evaluation results for the Times New Roman development set for different sliding window widths and overlaps. The results are reported in terms of character error rate (CER) which takes into account the errors due to insertion, deletion, and substitution. Based on the results, we applied a window width of six pixels with an overlap of three pixels for all other fonts. The error rates for the Thuluth font were significantly more than the mean error rates across other fonts, which prompted us to investigate possible explanations. We noticed that the text in Thuluth was very compact (and complex) compared with other fonts; thus, we decided to separately adjust the sliding window parameters for this font based on the recognition results for its development set. This finding indicates that feature extraction parameters may be separately optimized for each font if desired, which may lead to reduction in error rates. However, this situation will include overhead for the necessary time and resources for optimally configuring the parameters for each font.

Table 2: Sliding window's width and overlap based on evaluations of the development set using the Times New Roman font from the P-KHATT database.

Window (W|O)[*]	*No. of states*	*CER (%)*
4|2	10	1.43
2|0	11	1.78
3|1	10	2.54
1|0	17	2.60
3|0	7	1.65
4|1	7	1.37
5|2	7	1.26
6|3	**7**	**1.23**
4|0	6	2.32
6|2	6	1.87
8|4	5	1.56

[*] **W**: Width; **O**: Overlap

Once the sliding window parameters were selected, we performed the two-step training (i.e., uniform initialization and alignment-based initialization) for all eight fonts. The optimal number of states per HMM was determined based on the evaluation results for each font's development set. Character hypotheses for the development and evaluation sets were generated. Table 3 presents the recognition results for each of the eight fonts. The best result—CER 1.04%—was achieved for the Tahoma font. The worst result—CER 7.55%—was achieved for the Thuluth font. These results are understandable given that Tahoma is a simple font with wide-spaced characters, whereas Thuluth is a complex font with narrow character widths and numerous ligatures. The mean CER of 2.89% was achieved for the eight fonts on the evaluation sets. These results demonstrate the effectiveness of our adaptive sliding window technique considering our use of simple pixel density features with only six cells in a sliding window frame. We also report the statistical significance of the error rates at 95% confidence level i.e., in order for the error rates from another experiment on a particular set to be significantly different with 95% confidence, the difference in CER of the two experiments should be beyond the significance range.

Table 3: Results for mono-font text recognition using the P-KHATT database.

Font	*Window (W|O)*	*No. of States*	*CER (%)*		*Statistical Significance*
			Development	*Evaluation*	
Times New Roman	6|3	7	1.23	1.20	±0.06
Andalus	6|3	8	1.20	1.35	±0.07
DecoType Thuluth	**4|2**	7	7.51	**7.55**	±0.15
Tahoma	6|3	9	1.00	**1.04**	±0.06
Traditional Arabic	6|3	6	4.75	4.35	±0.12
Naskh	6|3	6	2.61	3.06	±0.10
Akbaar	6|3	6	2.80	2.87	±0.09
Simplified Arabic	6|3	7	2.02	1.67	±0.07
Mean			2.89	**2.89**	

4.1.3. Mixed-font Text Recognition

In this section, we present the experiment for mixed-font recognition. As an initial experiment, we trained a mixed-font recognizer (i.e., the recognizer was trained using training samples from all fonts). The training procedure was identical to the training procedure described in the previous section. The optimal HMM parameters were selected based on the results from the development set, and a final evaluation was conducted on the evaluation set. A CER of 12.19% for the development set and a CER of 12.14% for the evaluation sets were achieved, which are significantly higher than the mean CER of 2.89% that was achieved for mono-font text recognition. This increase in error rates can be partly explained by the large variation in font styles and the fact that each font has individual parameters (such as number of states), which is difficult to generalize. This motivated us to explore font-identification-based recognition, as described in Section 3.2.1, in which the image font is identified in the first step and the mono-font recognizer for the identified font is subsequently employed for text recognition in the second step.

For the font-identification-based recognition, we trained the font identification module. The font features described in Section 3.2.1 were extracted from the training samples for each of the eight fonts. A SVM with Radial Basis Function (RBF) as the kernel was employed as a classifier. The font identification module was evaluated using a set that contained 1,414 text line images for each font, which were randomly distributed. Table 4 presents the font identification results and the confusion matrix.

As shown in Table 4, we achieve reasonable results for the font identification, which demonstrates the effectiveness of our proposed features for font identification. Common confusion occurred between the Simplified Arabic font and the Times New Roman font. A closer look at the text images from the two fonts reveals that the two fonts are similar; this observation has been noted in other studies (c.f. [38]). Another observation is that both fonts employ the same number of HMM states per model, which provides clues regarding their similar properties. To confirm that the fonts are similar, we recognized the text images from the Times New Roman font using the mono-font recognizer that was trained on the Simplified Arabic font. A CER of 3.68% was achieved, which confirmed that the two fonts are not only visually similar but also exhibit similar properties with respect to text recognition. When they were combined as one font, our font identification rate increased to **97.27%.**

Table 4: Font identification results and the confusion matrix with the P-KHATT database.

Font	*AKH*	*AND*	*NAS*	*SIM*	*TAH*	*TLT*	*TNR*	*TRA*	***Identification Rate (%)***
AKH	1337	0	20	11	2	4	1	39	94.55
AND	2	1402	0	1	2	4	1	2	99.15
NAS	9	1	1352	1	0	20	2	29	95.62
SIM	5	1	14	1269	2	3	118	2	89.75
TAH	2	0	0	0	1405	4	1	2	99.36
TLT	2	1	26	2	0	1367	2	14	96.68
TNR	2	1	3	129	3	3	1272	1	89.96
TRA	14	1	33	3	0	8	3	1352	95.62
								Mean	**95.08**

After associating the font of the input text image, we performed feature extraction and recognition using the mono-font text recognizer of the associated font. With this approach, we achieved a CER of **3.44%**, which is closer to the mean CER that we achieved in the mono-font setups. This finding demonstrates the effectiveness of this approach compared with the commonly employed approach of recognizing the text image using a recognizer that is trained on multiple fonts. To understand the recognition errors caused by errors in font identification, we conducted another experiment, in which we manually separated the text line images based on the font and then recognized the text images using each font’s individual mono-font recognizer. The CER was 2.86%; thus, the text recognition error caused by the error in font identification was 0.58% (i.e., 3.44 - 2.86). Table 5 summarizes the results of the recognition for both scenarios.

Table 5: Results for the mixed-font text recognition experiments using the P-KHATT database.

Setup	*CER (%)*
Recognizer trained on samples from all fonts	12.19
Using font-association-based recognition	3.44
Recognition using the mono-font recognizer after manually separating text lines into different fonts.	2.86

4.1.4. Text Recognition of Unseen Fonts

In the last set of experiments, we performed text recognition on unseen fonts. We attempted different configurations to investigate the effectiveness of various approaches. In the first experiment, we recognized the unknown font's text images using the recognizer trained on text line images from all the eight fonts. In the second experiment, we associated the input text line images to the closest of the eight fonts using the font association module and subsequently employed the associated font's recognizer to recognize the input text. In the next few experiments, we evaluated the HMM adaptation technique and the font association step that was presented in Section 3.3.1. In one of the experiments, we employed 100 labeled text line images for the unseen font to perform MLLR-based supervised adaptation. Recognition was performed after the adaptation step. In another set of experiments, we investigated unsupervised HMM adaptation, in which no labeled data for the unseen fonts were employed. In the last two experiments, we applied character bigrams that were estimated from the training data as language models during the decoding step. The perplexity of the character bigrams on the evaluation set was 13.08. Character bigrams were employed with the supervised and unsupervised adaptations. A summary of the recognition results for the unseen fonts is presented in Table 6. Although the supervised and unsupervised adaptation techniques improve the results, the improvements based on the supervised adaptation are optimal, which is understandable. It assumes the availability of labeled samples for the input font, which may not always be feasible. The use of a language model further improves the results.

4.2. Text Recognition Using the APTI Database

In this section, we present the experiments that we conducted using the publicly available APTI database [6]. The main objective of the experiments was to demonstrate the robustness of our OCR system for printed Arabic text recognition in multiple databases.

Note that the characteristics of the APTI database differ from the P-KHATT database; as a result, are not comparable across the databases. In the following sections, we demonstrate that our OCR system can comparably perform with other systems that employ the same database, as reported in the literature.

Table 6: Text recognition results for the unseen font using the P-KHATT database.

Setup	*CER (%)*
Recognizer trained on samples from all fonts	19.28
Recognizer for the closest identified font	15.39
Recognizer for the closest identified font + Unsupervised adaptation	11.76
Recognizer for the closest identified font + Supervised adaptation	9.43
Recognizer for the closest identified font + Unsupervised adaptation + Character bigrams	8.39
Recognizer for the closest identified font + Supervised adaptation + Character bigrams	7.18

The APTI database is a publicly available database that is available at no cost for noncommercial use [6]. The database contains low-resolution (72 DPI) and synthetically generated printed Arabic word images in many fonts, sizes, and styles. The database is partitioned into six sets for each combination of font, size, and style. Five of the six sets are open, whereas the sixth set has not been disclosed to the public and is employed in competitions to evaluate submitted OCR systems. According to the database developers, the characteristics of the sixth set is similar to the characteristics of the remaining sets [6]. Each set contains different word images but the distribution of the characters is nearly identical in every set.

In the first set of experiments, we perform mono-font text recognition. We experimented with five different fonts from the APTI database; the same five fonts were selected in the first competition that was held using the APTI database [39]. For each font, we selected 24-point images in plain text. *Set-1* was selected as the training data, and 3000 images from *set-2* were selected as the development set to optimally configure the number of states per HMM. *Set-5* was utilized to evaluate the system’s performance. All images (with the exception of the images in the Diwani Letter font) were height-normalized to 64 pixels while maintaining a constant aspect ratio. Because Diwani Letter is very compact with many vertically overlapping ligatures, it was height-normalized to 96 pixels.

Features were extracted from the height-normalized images. Features from the training set and the image transcription were employed to train the HMMs for our continuous HMM system. Each character-shape model provided in the APTI database was applied as a separate HMM, and the different character shapes were mapped to the characters that they represented after the recognition. We employed *Bakis* topology for each model with a constant number of states per model, with the exception of the models that represented very narrow characters (e.g., *Alif*), which contain half the number of states. An explicit white space model was not employed in these experiments. The remaining details of the text recognition system are similar to the details of the experiments on P-KHATT database.

In Table 7, we present the mono-font text recognition results for the individual fonts. The text recognition results are presented in terms of the CER. Our text recognition results were acceptable for all fonts. The best results were obtained for the Arabic Transparent font, whereas the results for the Andalus and Simplified Arabic fonts were also reasonable. The poorest results were obtained for the Diwani Letter, which is a complex and compact font. The mean CER for all fonts was 2.07%. For optimal performance, parameters such as the sliding window width and overlap, the number of mixtures per HMM state, and the image height can be calibrated using the development set. Use of the font-specific ligature models also has the potential to improve recognition performance [24], [25].

Table 7: Results for mono-font text recognition using the APTI database.

Font	***CER (%)***
Andalus	0.76
Arabic Transparent	0.57
Diwani Letter	4.67
Simplified Arabic	0.69
Traditional Arabic	3.65
Mean	**2.07**

In Table 8, we compare our text recognition results using the APTI database with results from other HMM systems that have been reported in the literature using the APTI database. The comparison is based on recognition results on Arabic Transparent font because this font was included in the reference protocols for the text recognition competitions that employed the APTI database [39], [40]. For the remaining fonts, the results in the competitions are presented for mixed-font and multi-size text recognition scenarios. Thus, comparisons including other fonts are not possible. For the Arabic Transparent font, a completely objective comparison is still not possible for many reasons. One of the most important reasons is that *set-6* (not publicly available) was employed to evaluate the systems in the competitions. For other systems that utilized the APTI database and that are available in the literature, each group created individual training, development, and evaluation set partitions. Some systems applied word lexicons and n-gram language models, whereas other systems did not use any word lexicons or language models. For some systems, these details are not explicitly mentioned. Nevertheless, the comparison table can provide useful qualitative insights.

Table 8: Comparison with other HMM-based text recognition systems evaluated using the APTI database.

OCR systems	Database setup for experimentation	Error rates (%)	System description
UPV-PRHLT [39]	APTI database of printed Arabic text Font: Arabic Transparent, size: 24 - set 1 to set 5: used as training and development sets - set 6 (not publicly available) used for evaluation	Character level: 4.00 Word level: 15.60	- Bernoulli-mixture-based HMM system (BHMM)
Sameh and Khorsheed [41]	APTI database of printed Arabic text - Training Set: 80,000 images Sub-Training Set: 8,000 images Development Set: 1,000 images - Evaluation Set: 14,418 images	Character level: 3.35	- Discrete HMM-based OCR system - Sliding-window-based run-length encoding (RLE) features - Number of states per model and codebook size for feature quantization were optimized using the development set
IPSARec System [39]	APTI database of printed Arabic text Font: Arabic Transparent, size: 24 - sets 1 to set 5: used as training and development sets - set 6 (not publicly available) used for evaluation	Character level: 3.20 Word level: 22.50	- Discrete HMM-based OCR system - Pixel density features from the text image and its horizontal and vertical derivatives
THOCR1 [40]	APTI database of printed Arabic text Font: Arabic Transparent, size: 24 - set 1 to set 5: used as training and development sets - set 6 (not publicly available) used for evaluation	Character level: 1.05 Word level: 8.23	- HMM-based OCR system - Statistical and structural features and their derivatives - No language model used
THOCR2 [40]	APTI database of printed Arabic text Font: Arabic Transparent, size: 24 - sets 1 to set 5: used as training and development sets - set 6 (not publicly available) used for evaluation	Character level: 0.81 Word level: 4.97	- HMM-based OCR system - Statistical and structural features and their derivatives - Four-gram language model trained on the APTI training corpus used for rescoring

UPV-BHMM Khoury et al. [42]	APTI database of printed Arabic text. Font: Arabic Transparent, size: 24 - Training set: 10,000 images - Development set: 2000 images - Evaluation set: 3000 images	Character level: 0.30	- Bernoulli-mixture-based HMM system (BHMM) - Image height, sliding window width, number of states per model, and number of mixture components per state were optimized using the development set. - Five-gram language model at the character level
UPV-BHMM [40]	APTI database of printed Arabic text Font: Arabic Transparent, size: 24 - set 1 to set 5: used as training and development sets - set 6 (not publicly available) used for evaluation	Character level: 0.04 Word level: 0.10	- Character-based windowed BHMMs (Bernoulli HMMs) - Image height, sliding window width, number of states per model, and number of mixture components per state were optimized using the development set. - Five-gram language model at the character level
DIVA-REGIM [25], [39]	APTI database of printed Arabic text Font: Arabic Transparent, size: 24 - set 1 to set 5: used as training and development sets - set 6 (not publicly available) used for evaluation	Character level: 0.30 Word level: 1.10	- HMM based OCR system - Character shape as HMM models with some models merged into one model, which produced a total of 65 HMM models - Ergodic HMM topology with all possible transitions allowed - System parameters tuned using sets 1 to 5 - Number of connected black and white components, centers of gravity, density, compactness, vertical and horizontal - projection, baseline position, number of relative extrema in the vertical projection, and number of relative extrema in the horizontal projection used as features and their horizontal derivatives
SID [40]	APTI database of printed Arabic text Font: Arabic Transparent, size: 24 - set 1 to set 5: used as training and development sets set 6 (not publicly available) used for evaluation	Character level: 0.01 Word level: 2.59	- HMM-based OCR system - Sliding-window-based features
Present Work	APTI database of printed Arabic text Font: Arabic Transparent, size: 24 - Set 1 used as training set - Set 2 used as development set - Set 5 used as evaluation set	Character level: 0.57 Word level: 2.12	- HMM-based system with adaptive sliding window features and statistical feature in addition to its horizontal derivatives - Number of states per model optimized using the development set - Lexicon or language models not employed

In the next set of experiments, we performed mixed-font text recognition. Similar to the experiments with the P-KHATT database, we investigated two approaches to this text recognition task. The first approach was to train an HMM recognizer using samples from all five fonts. The second approach was to perform font-association-based recognition as presented in Section 3.2.1. For the first approach, we selected 3,000 word images from *set-1* of each font at 24-point size; the training data included 15,000 word images. The optimal number of states for the HMM was selected based on the recognition performance on the development set, which included 600 word images from *set-2* of each font (a total of 3,000 images). The final evaluation was conducted using the evaluation set, which included 15,000 word images in the five fonts (3,000 images from each font from *set-5*). A CER of 7.71% was obtained that was reasonable but higher than the mean CER of 2.07%, which was achieved in the mono-font setup.

For the second approach, we train our font-association module, which utilizes a SVM classifier with RBF kernel. The font identification features that are proposed in Section 3.2.1 were extracted from the 15,000 word images in the training set. These features and information about the word image font typefaces were employed to train the SVM classifier. The trained classifier was applied to associate the word image fonts in the evaluation set. The font identification results for the evaluation set are presented in Table 9. The font identification results are satisfactory; an average identification rate of 96.99% was obtained.

Table 9: Font identification results and the confusion matrix using the APTI database.

Font	Andalus	Arabic Transparent	Diwani Letter	Simplified Arabic	Traditional Arabic	***Identification Rate (%)***
Andalus	2994	2	0	3	1	99.80
Arabic Transparent	0	2806	12	178	4	93.53
Diwani Letter	0	4	2944	0	52	98.13
Simplified Arabic	0	140	0	2856	4	95.20
Traditional Arabic	3	3	44	2	2948	98.27
					Mean	**96.99**

After associating the input text image font, we perform feature extraction and recognition using the mono-font text recognizer for the associated font. Using this approach, we achieved a CER of **2.92%**, which demonstrates that the two-step font-association-based text recognition proved to be a better approach than performing text recognition trained on multiple font images. In Table 10, we summarize the text recognition results for the mixed-font text recognition task. When the text line images were manually separated based on the font, the CER was 2.12%. Consequently, the recognition error caused by the misclassified fonts was 0.80% (i.e., 2.92–2.12).

Table 10: Results for mixed-font text recognition using the APTI database.

Setup	*CER (%)*
Recognizer trained on samples from all fonts	7.71
Using font-association based recognition	2.92
Recognition using mono-font recognizer after manually separating text lines of different fonts.	2.12

5. Comparison with Similar Studies of Printed Arabic Text Recognition

In this section, we present a subjective comparison of our text recognition system with other HMM-based printed Arabic text recognition systems that have been discussed in the literature. Only studies that performed text recognition using text lines instead of systems that recognized isolated characters, digits, or word images were selected. Systems that employed synthetic databases were not selected because they did not address many of the practical challenges of real and scanned databases. In Table 11, we present a comparative study of different studies related to printed Arabic text line image recognition. This comparison was not performed to quantitatively compare different works because this task would be impossible due to the different databases utilized by different groups. Thus, this comparison should be understood from a complementary viewpoint. In the comparison presented in Table 11, we highlight different aspects of the study, such as the selected database, which was considered to be one of the most important aspects. The nature of the database, its text sources, its characteristics (such as scanning resolution and noise level), and its division into different sets (for training, development, and evaluation) serve an important role in text recognition performance.

Another important aspect is the nature of text recognition with respect to font variability. Some studies only reported their results for mono-font or mixed-font text recognition, whereas other studies discussed the performance of both mono-font and mixed-font text recognition. This study focuses on mono-font and mixed-font text recognition, as well as text recognition of unseen fonts. Other important aspects include the decoding network and the use of language models. Some studies optionally decode at the character level using character n-grams as their language models. Other studies have employed word lexicons with the optional use of word n-grams as language models. The issue of out-of-vocabulary (OOV) words is important when using word lexicons in open vocabulary word recognition tasks. One study (Prasad et al. [8]) also investigated the use of parts of Arabic words (PAW) language models. These models can also be used after decoding to re-score the N-best list that is generated during decoding.

Table 11: Subjective comparison of other HMM-based printed Arabic text recognition systems that perform recognition at the text line level

Work	Characteristics of the database	Main aspects of text recognition	System description	Error rates (%)
Bazzi et al. [4]	DARPA Arabic OCR Corpus of 345 pages of Arabic text scanned at 600 DPI For mixed font text recognition: • Text line images from 30 pages were used for training • Text line images from 10 pages were used for evaluation	• Mono-font text recognition • Mixed-font text recognition where the training set and the evaluation set contains line images from four different fonts	• HMM-based OCR system • Pixels density features with vertical and horizontal derivatives in addition to local slope and correlation features across a window of two cells • lexicon obtained from a large text corpus with closed vocabulary of 30k words • A language model for recognition from the same text corpus	CER of 0.40 for mono-font text recognition CER of 2.60 for mixed-font system with closed vocabulary word recognition CER of 4.50 on mixed font open vocabulary text recognition using trigram character language model
Natarajan et al. [21]	DARPA Arabic OCR Corpus of 345 pages of Arabic text scanned at 600 DPI • Text line images from 192 text zones were used for training • Text line images from 102 text zones were used for evaluation	• Mixed-font text recognition	• HMM-based OCR system with mixture tying at character level • Percentile features with vertical and horizontal derivatives in addition to local slope and correlation features	CER of 3.86
Khorsheed [20]	A database of 15,000 text line images in six different fonts i.e. 2,500 text line images in each font. • Training set includes 1,500 text line images in each of the six font • Development Set includes 1,000 text line images in each of the six font	• Mono-font text recognition for six different fonts	• Discrete-HMMs based OCR system • Pixel density features extracted from the sliding windows over the text line images and their horizontal and vertical derivatives • Contextual HMM modeling • Character bigrams from training transcriptions	CER ranging from 7.40 (for *Andalus* font) to 14.00 (for *Naskh* font)

Prasad et al. [8]	DARPA Arabic Machine Print (DAMP) scanned at 600 DPI • Training set includes text line images from 177 page images in addition to text line images from 380 synthetically generated page images in multiple fonts and sizes • Development set includes text line images from 60 page images • Evaluation set includes text line images from 60 page images	• Mixed-font text recognition	• HMM-based OCR system with discriminative training • Position-dependent tied mixtures where the Gaussians for corresponding states of all the presentation forms of character is tied • Contextual HMM modeling • Character, PAW, and word trigrams from 2.6 million words of Arabic newswire data in addition to the training transcriptions • Word lexicon of 65k words	Best word error rate of 9.60 using PAW language model and N-Best rescoring using contextual HMMs estimated using discriminative training procedure
Dreuw et al. [7]	RAMP-N printed Arabic database in 20 different fonts scanned at 600 DPI: • 222,421 text line images for training • 1,155 text line images for the development set • 3,480 text line images for the evaluation set	• Mixed-font text recognition (two of the fonts cover more than 95% of all the text line images in the evaluation set) • Word recognition task with Out Of Vocabulary rate of 2.21%	• HMM-based system with ML trained GMMs with globally pooled variances • Appearance-based image slice features along with spatial derivatives • Language model using a corpus of 228 million running words • Vocabulary size of 106k words	WER of 4.76 and CER of 0.15 on the rendered data WER of 5.79 and CER of 0.66 on the scanned data
Present Work	P-KHATT printed Arabic text database in eight different fonts scanned at 300 DPI: • Training set includes 6,472 text line images in each of the eight font (2,000 text line images used for training in current work) • Development Set includes 1,414 text line images in each of the eight font • Evaluation Set includes 1,424 text line images in each of the eight font	• Mono-font text recognition • Mixed-font text recognition • Text recognition for unseen font (i.e., having no training samples)	• HMM-based OCR system • Adaptive sliding window for feature extraction • Pixel density features and its vertical and horizontal derivatives • Font identification based text recognition • Use of supervised and unsupervised HMM adaption techniques to deal with font variability • Character bigrams from training transcriptions	CER ranging from 1.04 (for Tahoma) to 7.55 (for Thuluth) for mono-font text recognition without using any language model or word lexicon CER of 3.44 for mixed-font text recognition without using any language model or word lexicon For unseen-font text recognition: CER of 11.76 using unsupervised adaptation without any language models and lexicon CER 7.18 using supervised adaptation and character bigrams as language model

In addition to these aspects, other aspects can be compared between different studies, including the nature of the HMM system (continuous vs. discrete vs. systems with differing levels of tying, e.g., mixture tying and state tying), the sliding window technique and features employed for recognition.

6. Conclusions

Text recognition is an active area of research. Although printed text recognition is well researched and developed compared with handwritten text recognition, recognition in mixed-font scenarios or for unseen fonts remains challenging. This paper presents HMM-based printed Arabic text recognition in different scenarios. A novel method for adaptively dividing the sliding window into cells was presented. The technique utilizes the writing line property of Arabic text and ink-pixel distributions. Simple pixel density features using the proposed adaptive window yielded reasonable text recognition results.

For mixed-font and unseen-font text recognition, we employ a two-step approach, in which the input text line image is associated with the closest known font in the first step and the HMM-based text recognition is performed in the second step using the recognizer that was trained on the associated font's text. This approach is more effective than the common approach of recognizing the text using a recognizer that was trained on samples of different fonts. To associate the input text with a known font, we presented simple and effective features for font identification. Experiments that were conducted using the proposed features yielded acceptable font identification results for an evaluation set that contained eight commonly used fonts.

We also investigated the use of MLLR-based HMM adaptation techniques for text recognition in the unseen-font scenarios. We experimented with both supervised and unsupervised adaptation techniques. Supervised adaptation proved to be more effective compared with unsupervised adaptation but required a few samples of labeled data in the unseen font. Conversely, unsupervised adaptation required no labeled data and achieved better recognition results. In this paper, we presented a framework for printed text recognition that integrated different phases. A new database of printed Arabic text (the P-KHATT) in eight different fonts was introduced and employed for evaluations. The database will be made available to the research community at no cost. We also experimented with the publicly available APTI database of printed Arabic words in multiple fonts. The APTI database results are comparable with the HMM systems that have been evaluated using the same database. We also presented a qualitative comparison of our study with similar studies of printed Arabic text recognition using HMMs.

Although considerably high results can be achieved for printed text recognition, many challenges need to be addressed. The recognition of text from multiple unseen fonts may present issues given that HMM adaptations may not be particularly effective in these scenarios. Text recognition with highly complex fonts and degraded documents is difficult and remains an open area of research. An investigation of the use of post-processing techniques, such as spell correction, to enhance recognition rates may prove interesting.

Acknowledgement

The authors would like to acknowledge the support provided by King Fahd University of Petroleum and Minerals (KFUPM) for funding this work through project number RG 1313-1/2.

References

[1] B. Al-Badr and S. A. Mahmoud, "Survey and bibliography of Arabic optical text recognition," *Signal Processing*, vol. 41, no. 1, pp. 49–77, 1995.

[2] G. A. Fink, *Markov Models for Pattern Recognition*, 2nd ed. London: Springer London, 2014.

[3] T. Plötz and G. A. Fink, "Markov models for offline handwriting recognition: a survey," *Int. J. Doc. Anal. Recognit.*, vol. 12, no. 4, pp. 269–298, Oct. 2009.

[4] I. Bazzi, R. Schwartz, and J. Makhoul, "An omnifont open-vocabulary OCR system for English and Arabic," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 21, no. 6, pp. 495–504, Jun. 1999.

[5] M. P. Schambach, J. Rottland, and T. Alary, "How to convert a Latin handwriting recognition system to Arabic," in *Proc. of Eleventh International Conference on Frontiers in Handwriting Recognition*, 2008, pp. 265–270.

[6] F. Slimane, R. Ingold, S. Kanoun, A. M. Alimi, and J. Hennebert, "A New Arabic Printed Text Image Database and Evaluation Protocols," in *10th International Conference on Document Analysis and Recognition*, 2009, pp. 946–950.

[7] P. Dreuw, D. Rybach, G. Heigold, and H. Ney, "RWTH OCR: A Large Vocabulary Optical Character Recognition System for Arabic Scripts," in *Guide to OCR for Arabic Scripts SE - 9*, V. Märgner and H. El Abed, Eds. Springer London, 2012, pp. 215–254.

[8] R. Prasad, S. Saleem, M. Kamali, R. Meermeier, and P. Natarajan, "Improvements in hidden Markov model based Arabic OCR," in *Proc. of 19th International Conference on Pattern Recognition*, 2008, pp. 1–4.

[9] K. Ait-Mohand, T. Paquet, and N. Ragot, "Combining structure and parameter adaptation of HMMs for printed text recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2014.

[10] J. Mantas, "An overview of character recognition methodologies," *Pattern Recognit.*, vol. 19, no. 6, pp. 425–430, Jan. 1986.

[11] I. Marosi, "Industrial OCR approaches: architecture, algorithms, and adaptation techniques," in *Proc. of SPIE, Document Recognition and Retrieval XIV*, 2007, vol. 6500, pp. 650002–650010.

[12] Y.-Y. Chiang and C. A. Knoblock, "Recognition of Multi-oriented, Multi-sized, and Curved Text," in *Proc. of Eleventh International Conference on Document Analysis and Recognition*, 2011, pp. 1399–1403.

[13] V. Märgner and H. El Abed, Eds., *Guide to OCR for Arabic Scripts*. London: Springer London, 2012.

[14] S. Impedovo, L. Ottaviano, and S. Occhinegro, "Optical Character Recognition — A Survey," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 05, no. 01n02, pp. 1–24, Jun. 1991.

[15] Q. Tian, P. Zhang, T. Alexander, and Y. Kim, "Survey: Omnifont printed character recognition," *Vis. Commun. Image Process Image Process*, pp. 260–268, 1991.

[16] J. Trenkle, A. Gillies, E. Erlandson, S. Schlosser, and S. Cavin, "Advances in Arabic text recognition," in *Proc. Symp. Document Image Understanding Technology*, 2001.

[17] M. S. Khorsheed, "Off-line Arabic character recognition--a review," *Pattern Anal. Appl.*, vol. 5, no. 1, pp. 31–45, 2002.

[18] N. Arica and F. T. Yarman-Vural, "An overview of character recognition focused on off-line handwriting," *IEEE Trans. Syst. Man Cybern. Part C (Applications Rev.*, vol. 31, no. 2, pp. 216–233, May 2001.

[19] A. Amin, "Off-line Arabic character recognition: the state of the art," *Pattern Recognit.*, vol. 31, no. 5, pp. 517–530, Mar. 1998.

[20] M. S. Khorsheed, "Offline recognition of omnifont Arabic text using the HMM ToolKit (HTK)," *Pattern Recognit. Lett.*, vol. 28, no. 12, pp. 1563–1571, Sep. 2007.

[21] P. Natrajan, Z. Lu, R. Schwartz, I. Bazzi, and J. Makhoul, "Multilingual Machine Printed OCR," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 15, no. 01, pp. 43–63, Feb. 2001.

[22] H. A. Al-Muhtaseb, S. A. Mahmoud, and R. S. Qahwaji, "Recognition of off-line printed Arabic text using Hidden Markov Models," *Signal Processing*, vol. 88, no. 12, pp. 2902–2912, 2008.

[23] I. Ahmad, S. A. Mahmoud, and M. T. Parvez, "Printed Arabic Text Recognition," in *Guide to OCR for Arabic Scripts*, V. Märgner and H. El Abed, Eds. Springer London, 2012, pp. 147–168.

[24] F. Slimane, O. Zayene, S. Kanoun, A. Alimi, J. Hennebert, and R. Ingold, "New features for complex Arabic fonts in cascading recognition system," in *Proc. of 21st International Conference on Pattern Recognition*, 2012, pp. 738–741.

[25] F. Slimane, R. Ingold, S. Kanoun, A. M. Alimi, and J. Hennebert, "Impact of Character Models Choice on Arabic Text Recognition Performance," in *2010 12th International Conference on Frontiers in Handwriting Recognition*, 2010, pp. 670–675.

[26] I. Ahmad, L. Rothacker, G. A. Fink, and S. A. Mahmoud, "Novel Sub-character HMM Models for Arabic Text Recognition," in *2013 12th International Conference on Document Analysis and Recognition*, 2013, pp. 658–662.

[27] I. Ahmad, G. A. Fink, and S. A. Mahmoud, "Improvements in Sub-Character HMM Model Based Arabic Text Recognition," in *14th International Conference on Frontiers in Handwriting Recognition*, 2014, pp. 537–542.

[28] S. J. Young, G. Evermann, T. Hain, D. Kershaw, G. Moore, J. Odell, D. Ollason, D. Povey, V. Valtchev, and P. Woodland, *The HTK Book (for HTK Version 3.2. 1)*. Cambridge University Engineering Department, 2002.

[29] U.-V. Marti and H. Bunke, "Handwritten sentence recognition," in *Proc. of 15th International Conference on Pattern Recognition. ICPR-2000*, 2000, pp. 463–466.

[30] S. Saleem, H. Cao, K. Subramanian, M. Kamali, R. Prasad, and P. Natarajan, "Improvements in BBN's HMM-based offline Arabic handwriting recognition system," in *Proc. of 10th International Conference on Document Analysis and Recognition, ICDAR'09*, 2009, pp. 773–777.

[31] R. Al-Hajj Mohamad, L. Likforman-Sulem, and C. Mokbel, "Combining slanted-frame classifiers for improved HMM-based Arabic handwriting recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 7, pp. 1165–1177, 2009.

[32] I. Bazzi, C. LaPre, J. Makhoul, C. Raphael, and R. Schwartz, "Omnifont and unlimited-vocabulary OCR for English and Arabic," in *Proc. of the Fourth International Conference on Document Analysis and Recognition*, 1997, vol. 2, pp. 842 –846.

[33] M. J. F. Gales and P. C. Woodland, "Mean and variance adaptation within the MLLR framework," *Comput. Speech Lang.*, vol. 10, no. 4, pp. 249–264, Oct. 1996.

[34] C. J. Leggetter and P. C. Woodland, "Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models," *Comput. Speech Lang.*, vol. 9, no. 2, pp. 171–185, Apr. 1995.

[35] S. A. Mahmoud, I. Ahmad, M. Alshayeb, and W. G. Al-Khatib, "A Database for Offline Arabic Handwritten Text Recognition," *Image Anal. Recognit.*, pp. 397–406, 2011.

[36] S. A. Mahmoud, I. Ahmad, W. G. Al-Khatib, M. Alshayeb, M. Tanvir Parvez, V. Märgner, and G. A. Fink, "KHATT: An open Arabic offline handwritten text database," *Pattern Recognit.*, vol. 47, no. 3, pp. 1096–1112, Mar. 2014.

[37] I. Ahmad, “A Technique for Skew Detection of Printed Arabic Documents,” in *Proc. of 10th International Conference Computer Graphics, Imaging and Visualization (CGIV)*, 2013, pp. 62–67.

[38] H. Luqman, S. A. Mahmoud, and S. Awaida, “KAFD Arabic font database,” *Pattern Recognit.*, vol. 47, no. 6, pp. 2231–2240, Jun. 2014.

[39] F. Slimane, S. Kanoun, H. El Abed, A. M. Alimi, R. Ingold, and J. Hennebert, “ICDAR 2011 - Arabic Recognition Competition: Multi-font Multi-size Digitally Represented Text,” in *2011 International Conference on Document Analysis and Recognition*, 2011, pp. 1449–1453.

[40] F. Slimane, S. Kanoun, H. El Abed, A. M. Alimi, R. Ingold, and J. Hennebert, “ICDAR2013 Competition on Multi-font and Multi-size Digitally Represented Arabic Text,” in *2013 12th International Conference on Document Analysis and Recognition*, 2013, pp. 1433–1437.

[41] S. M. Awaida and M. S. Khorsheed, “Developing discrete density Hidden Markov Models for Arabic printed text recognition,” in *2012 IEEE International Conference on Computational Intelligence and Cybernetics (CyberneticsCom)*, 2012, pp. 35–39.

[42] I. Khoury, A. Giménez, A. Juan, and J. Andrés-Ferrer, “Arabic Printed Word Recognition Using Windowed Bernoulli HMMs,” in *Image Analysis and Processing – ICIAP 2013 SE - 34*, vol. 8156, A. Petrosino, Ed. Springer Berlin Heidelberg, 2013, pp. 330–339.