Umm Al-Qura Univ. J. Eng.& Arch. Vol. 2, No.2, pp. 53-65 (2010) 30-11-009

A Fully Automated Image Database Creation Tool For Developing Pattern Recognition Systems

Khalid Abdulhameed Al-Hindi

خالد عبدالحميد الهندي

Computer Engineering Department, Faculty of Computer and Information Systems, Umm AL-Qura University, Saudi Arabia
kahindi@uqu.edu.sa

أداة لإنشاء قواعد بيانات الصور المعدة لتطوير أنظمة التعرف على الأنماط

يتطلب تطوير أنظمة التعرف الذكية توفر قاعدة بيانات من النماذج والتي ستستخدم حصرا لإنشاء تلك الأنظمة واختبارها ومن ثم التحقق من فعاليتها. وتشمل تلك الأنظمة مثلا التعرف على الحروف ضوئيا ، والتعرف على بصمات الأصابع ، والتعرف على الأوجه . ويعمد المطورون والباحثون عادة إلى تجميع تلك البيانات يدويا وذلك برقمنة صور النماذج باستخدام الماسح الضوئي ثم البدء بتحليل كل صورة على حدة وتحديد كل نموذج وحفظه بشكل مستقل. وتأخذ كل هذه العمليات الوقت والجهد الكبيرين . ويقدم هذا البحث أداة تجعل من عملية إنشاء قواعد بيانات النماذج أكثر سهولة وسرعة بتوفير استمارة مخصصة لذلك ، وهذه الاستمارة توائم العديد من التطبيقات مثل بصمات الأصابع والتواقيع الشخصية وخط اليد . وبعد الانتهاء من تجميع الاستمارات المعبأة بالبيانات يتم تمريرها في أداة البرمجة باستخدام الماتلاب والتي تحوي واجهة المستخدم الرسومية المتصلة بالماسح الضوئي لرقمنة الاستمارات وفصل النماذج فيها آليا. لذا استبدلت أداة البرمجة العديد من مراحل إنشاء قواعد البيانات والتي تتم عادة يدويا بخطوات أخرى آلية ، وهو ما سيوفر الكثير من الوقت والجهد . محليا، يمكن أن تشجع هذه الأداة لإنشاء المزيد من قواعد البيانات الخاصة بالتطبيقات العربية لأنظمة التعرف الذكية والتي لاقت القليل من الاهتمام بالمقارنة بمثيلاتها الإنجليزية.

ABSTRACT

Developing intelligent recognition systems requires a database of samples to be used extensively for constructing, testing, and verification. Such systems include optical character recognition, fingerprint recognition, and face recognition. Developers and researchers tend to collect databases manually by digitizing the collected filled forms using available scanners and storing their digital images. Then every individual image is processed separately to specify regions of interest that are extracted, labeled, and stored to database. All mentioned steps are usually performed manually consuming much time and effort. This paper introduces a tool that will help researchers to speedily acquire sample prototypes using a general form. The form is designed to accommodate a wider range of applications such as fingerprints, signatures, and handwritings. Filled forms are processed by a Matlab-based (GUI) tool that can automatically communicate with the scanner to digitize forms and separate prototypes into sub-images. Thus the tool saves a lot of time and effort for database creation by replacing several database creation stages that are usually performed manually with simple automated stages. Locally the tool is expected to encourage researchers to build databases for Arabic intelligent recognition applications that have received less attention if compared with English.

Key words: Intelligent recognition system, handwriting recognition, database creation, optical character recognition, document analysis

1. INTRODUCTION

The formation of a database is the initial stage in the development of any intelligent recognition system. In handwriting recognition applications, for instance, thousands of prototype samples need to be collected for developing and testing a reliable identification system. Databases for English handwriting recognition have been publicly available for a long time. Examples of such databases include NIST, CEDAR, and USA Zip Code. In contrast a large number of studies have reported that available databases for Arabic recognition applications are still very limited if compared with English language (Slimane, et. al. 2009; Lorigo and Govindaraju 2006; Abdel Raouf, Higgins, and Khalil 2008; Al Shalabi, Hasan, and Ali 2005; Saabni and El-Sana 2009). In the case of Arabic handwriting recognition, for example, researchers tend to use small and specific databases. Sometimes large databases do exist but with no access to the public (Al-Ohali, Cheriet, and Suen 2003). Thus the need for constructing huge databases for different Arabic language recognition applications is crucial. Accordingly, this study will focus on Arabic intelligent recognition applications that have received less attention in literature.

Current studies have applied different approaches to database creation. (The first) group of studies have created database totally manually. For example, the database of handwritten Arabic words, numbers, and signatures collected by Kharma et al. and Ma'adeed et al. (Al-Ma'adeed, Elliman, and Higgins 2004; Kharma, Ahmed, and Ward, R. 1999) were totally done manually. The filled forms were digitized using scanners and their digital images where stored. Then every individual image was processed separately to specify areas of interest that were extracted, labeled, and added to database. All mentioned steps were performed manually that consumed a great deal of time and effort. As a result this approach is not favorable.

Another group of researchers constructed a special form for database creation with special features. These features were used later for extracting prototypes using some digital image processing techniques. For example, in the IFN/ENIT database (Pechwitz et. al. 2002; Pechwitz and Maergner 2003; El Abed and Margner 2007) for collecting Arabic handwritings a special form was designed. Digitizing collected forms were done manually at the first stage. Then sub-images of words and postcodes are extracted from every form image using a specially developed function based on digital image processing computations. Forms were limited to handwritings and could not be generalized to be used for other types of data such as fingerprints. Thus, this approach is also time consuming and limited to the designated purpose.

One important note is that most cited papers have used flatbed scanners during the digitization process of filled forms. This technique requires additional time and effort to handle manually each form and opening scanning application and storing images. Alternatively utilizing a sheet-fed scanner and programmatically controlling the communication with its driver would be a better solution to speed up the process of database creation. Note that all these steps could be eliminated if a tool is available to researchers.

This research has been mainly motivated by the lack of an easy tool that directly acquires image from scanner and then manages storing samples into database. Such a system will provide researchers with a fast and easy tool so that they can build huge databases for intelligent recognition systems. Thus this paper will address those issues by developing a tool with such features. Locally the tool is expected to encourage researchers to build databases for Arabic recognition applications such as handwriting recognition and printed letter recognition.

2. FORM DESIGN

A tool for collecting databases for the development of artificial recognition systems is introduced in this paper. It consists of two components: a form for collecting data and a program for digitizing and extracting prototypes. This section deals with the design issues of the form. Section 3 describes the process of extracting prototype samples from a digitized form using some digital image processing algorithms. The development of a graphical user interface (GUI) for the tool and the interfacing with the scanner are explained in the next section. Later sections describe tool testing and conclusions.

The database collection form is designed in this paper taking into account the following characteristics. First, form must be easily understood by users who are required to fill in. Second, form image must have special features so that it could be automatically processed using various digital image processing techniques at high speed. Third, writing cells must be adequate to include different types of data such as fingerprints or personal signatures. Fourth, form provides additional information about the filling person to be used by researches.

Based on the previous requirements, the form shown in Figure 1 is introduced. It is filled with sample personal signature data although it can be used for collecting other types of data such as fingerprints and handwritings. The form consists of two areas. The upper area of the form is reserved for

personal information. The lower area , which is divided into 30 small areas(,) is for collecting prototypes. Each area is measured (at) 1.2 by 1.2 inch approximately to accommodate different types of images such as fingerprints, signatures, and handwritings. Thus, the form can be used to construct and test different artificial recognition applications.

Form is created as a black-and-white Latex document of A4 size. Latex is preferred over other known text processing programs such as Microsoft Word due to its simple editing and formatting operations. Latex produces a form in device independent (DVI) file format that requires special viewers installed in the computer system. Since many users have no access to a DVI viewer, another version is generated in the well-known portable document file (PDF) format using pdfLatex. The form must be printed on a white A4 paper using black ink (LaserJet or inkjet) to be digitally processed on a later stage.

3. EXTRACTION OF INDIVIDUAL PROTOTYPES

In this paper, a Matlab function named form_read is designed to automatically process form images in order to extract prototype into separate image files. It utilizes a few techniques offered by digital image processing computing to achieve goals (Gonzalez, Woods, and Eddins 2003; Snyder and Qi 2004; Nixon and Aguado 2008). Figure 2 depicts the flowchart of the developed function. As the user accesses the main GUI, the input image color depth and resolution must be determined first. Color depth configurations include color, grayscale, and binary (black and white).

The designed function processes a form image based on its black and white color depth variety. Thus(,) if needed(,) color images are converted into grayscale by simply taking the average of the three color channels: red, green, and blue (RGB). If a form image is a grayscale then it is converted into binary using a global thresholding level determined by Otsu's method (Otsu 1979).

After obtaining a binary image for the scanned form, the horizontal and vertical projection profiles are computed and carefully analyzed. The function employs the distinct characteristics of horizontal and vertical projection profiles shown in Figure 3 to locate image separation points. High projection amplitudes are attained at table gridlines (Jain, Kasturi, and Shunck 1995) provided that the form is scanned exactly at right angle. However during scanning operation, scanner roller mechanism causes forms to be slightly rotated, which results in a skewed image as shown in Figure 4. Accordingly, a script is added to the function to estimate skew angle and use the information to precisely adjust corner points of prototypes. It is simply based on slightly

rotating the image in both directions until the horizontal and vertical projections are equal in length with actual table gridlines. Experimental testing shows that a calibration angle in range $\theta \in (-1.0, +1.0^o)$ is adequate for the designed system.

Finally prototype images are extracted based on located prototype corners. Extra white areas surrounding a prototype are automatically detected and removed using projection profile analysis. Prototypes are then stored as bitmap images of the chosen color depth. Every form produces 30 different prototypes that belong to a particular participant. Image file naming is set to take the format n_m_o.bmp, where *n* is the form number, *m* is the row number ($m \in \{1,2,\ldots,6\}$), and *o* is the column number ($o \in \{1,2,\ldots,5\}$).

4. FORM DIGITIZATION AND GUI DESIGN

The formation of image databases requires the digitization of the collected filled forms. This process can be achieved using scanners preferably with sheet-fed capability to speed up the acquisition process. The tool designed in this paper employs a DR-2080C Canon automatic-feed scanner that can automatically scan batches of documents up to 50 pages. It also offers a maximum resolution of 600 dots per inch (dpi) for 24-bit color, 8-bit grayscale, or 1-bit black-and-white images. Its compatibility with TWAIN standard driver facilitates the integration with the tool.

A graphical user interface (GUI) is designed to facilitate easy communication with the scanner and to initiate the form processing operation. The design is carried out using Matlab because it provides an easy GUI development tool called GUIDE (Graphical User Interface Development Environment) (Smith 2006). Figure 5 shows the GUI control panel for the tool. Note that the GUI design is intentionally kept simple with a minimal number of keys to speed up the database collection process.

The operation of this tool starts with loading the sheet-fed scanner with a batch of the collected filled forms. Then from the menu shown in Figure 5, users are able to configure the two scanning parameters: resolution in dpi and color depth. Color depth configurations include color, grayscale, and binary (black and white). Hitting the "Start Form Processing From Scanner" button causes the tool to start communicating with the scanner driver and acquire form images. Images are stored in the directory specified by the user in the "Directory Setting" panel. Users can also process form images that had been scanned independently by hitting the "Start Form Processing From Directory" button.

Interfacing the scanner with the tool requires a special program that provides communication features between a TWAIN scanner and the GUI. In this paper Quickscan is utilized to communicate with TWAIN scanners and convert the scanned form into a bitmap .bmp image. Quickscan is a suitable solution for developers looking for programmatically integrating the scanning process with a GUI. It is a command line TWAIN scanning utility that can be configured either by command line parameters or a configuration file. Optional papermakers include resolution, page size, output format, scanner choice, etc.

Form images are stored in Microsoft bitmap .bmp image file format. This format is preferred for two reasons. First, is it standard for any Windows-based operating system. Second, image data can be stored with lossless compression to prevent unnecessary altering of actual scanned data.

5. SYSTEM TESTING

In order to test the designed tool for database creation, a large database of signatures was collected. 70 forms were distributed to 70 different student participants. They were asked to freely jot their signatures 30 times as they may normally occur. Thus the total number of collected signatures is 2100 (70×30). Filled forms are processed using the tool and their images are used to calibrate form_read function until a %100 performance is achieved. Most of the performed adjustments are due to skewed images during scanning. The tool finally gives the desired performance.

6. CONCLUSIONS

A tool for collecting databases for the development of artificial recognition systems has been introduced. The tool consists of two components: a form for collecting data and a program for digitizing and extracting prototypes. The form is designed to contain thirty 1.2 square inch areas for sample data. It can be used for different types of recognition applications such as fingerprints, signatures, and handwritings. The design of accompanied program is based on Matlab that provides a GUI design environment and digital image processing functions. The tool enables researchers to directly digitize filled database forms by calling a sheet-fed scanner with TWAIN capability. Then it automatically locates the 30 prototypes on the form using projection profile analysis of digital image processing. The prototypes are finally stored and labeled in separate files. Thus the tool saves a lot of time and effort for database creation by replacing several database creation stages that are usually performed manually with simple automated stages.

Database Collection Form

التاريخ	١٤٢٧/٤/٢٥ هـ
المرحلة التعليمية	ابتدائي متوسط ثانوي (بكالوريوس) ماجستير دكتوراة
العمر	٢٨
النوع	ذكر (أنثى)
رقم المشارك	١

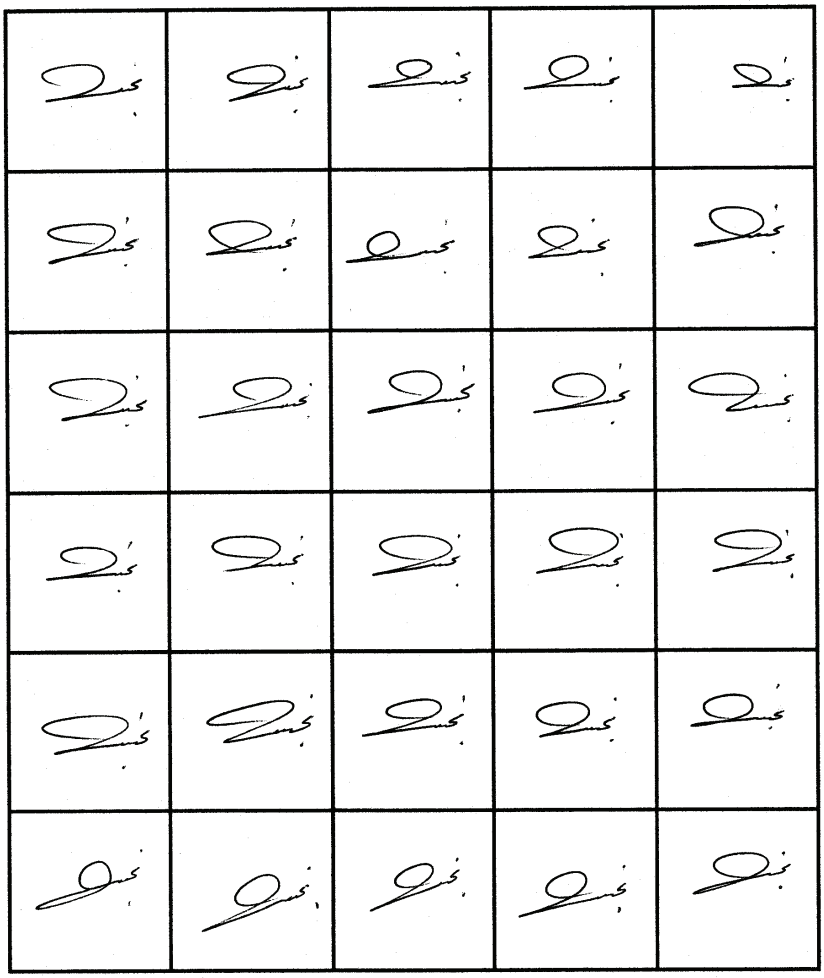

Figure 1. The designed form for database creation filled with example data.

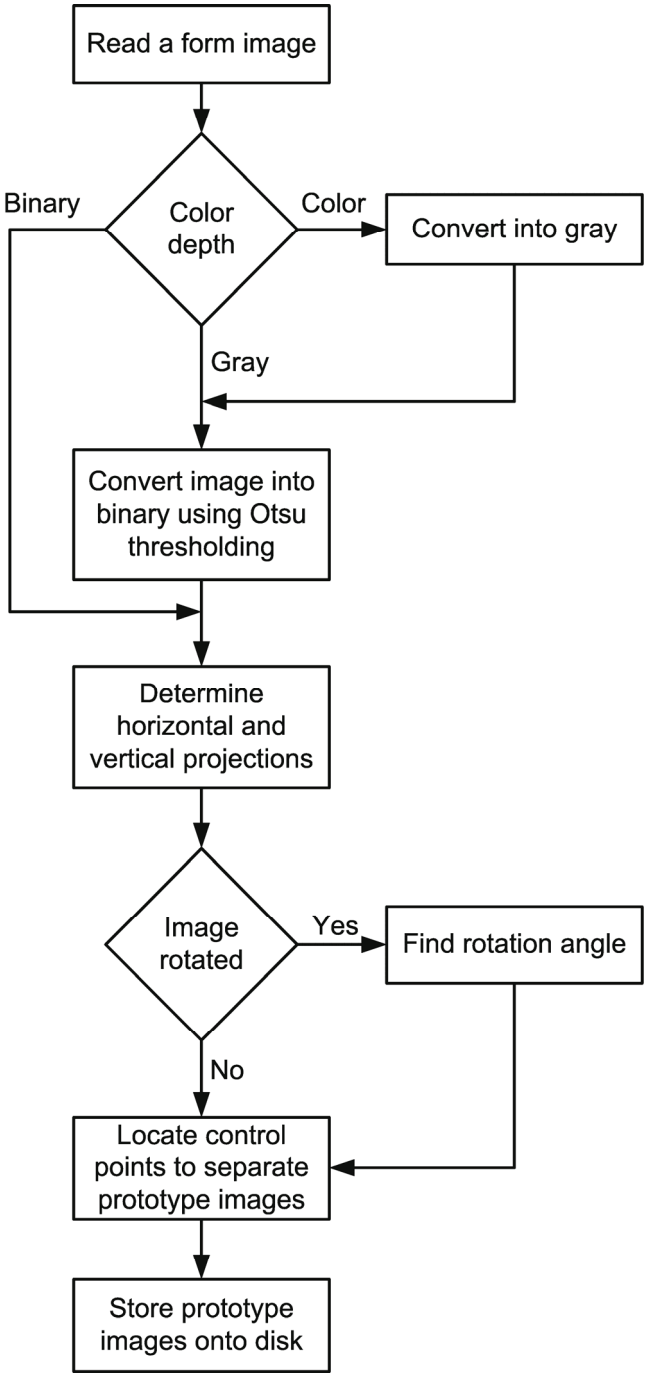

Figure 2. Flowchart of the developed form processing function.

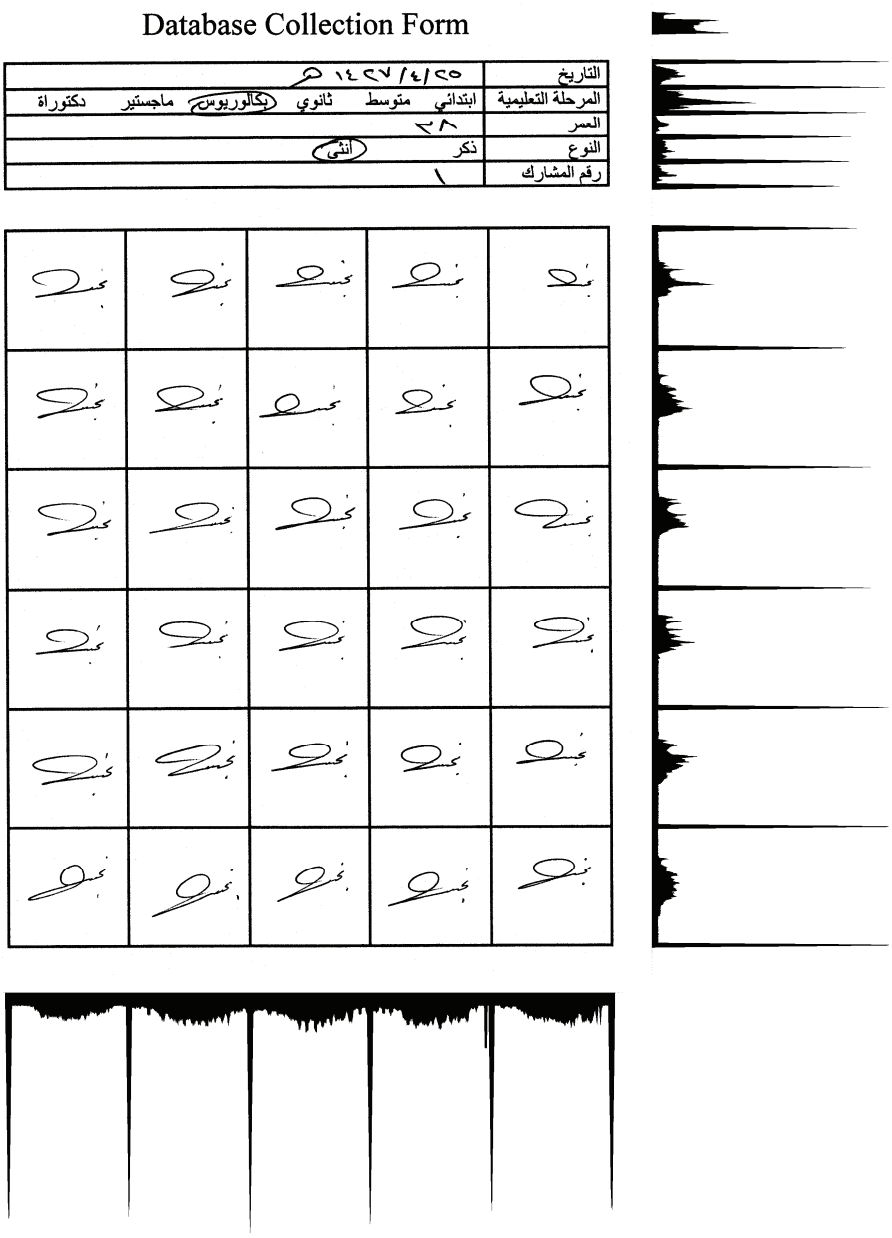

Figure 3. The distinct horizontal and vertical projection characteristics of form table gridlines are used to locate separation points of every prototype.

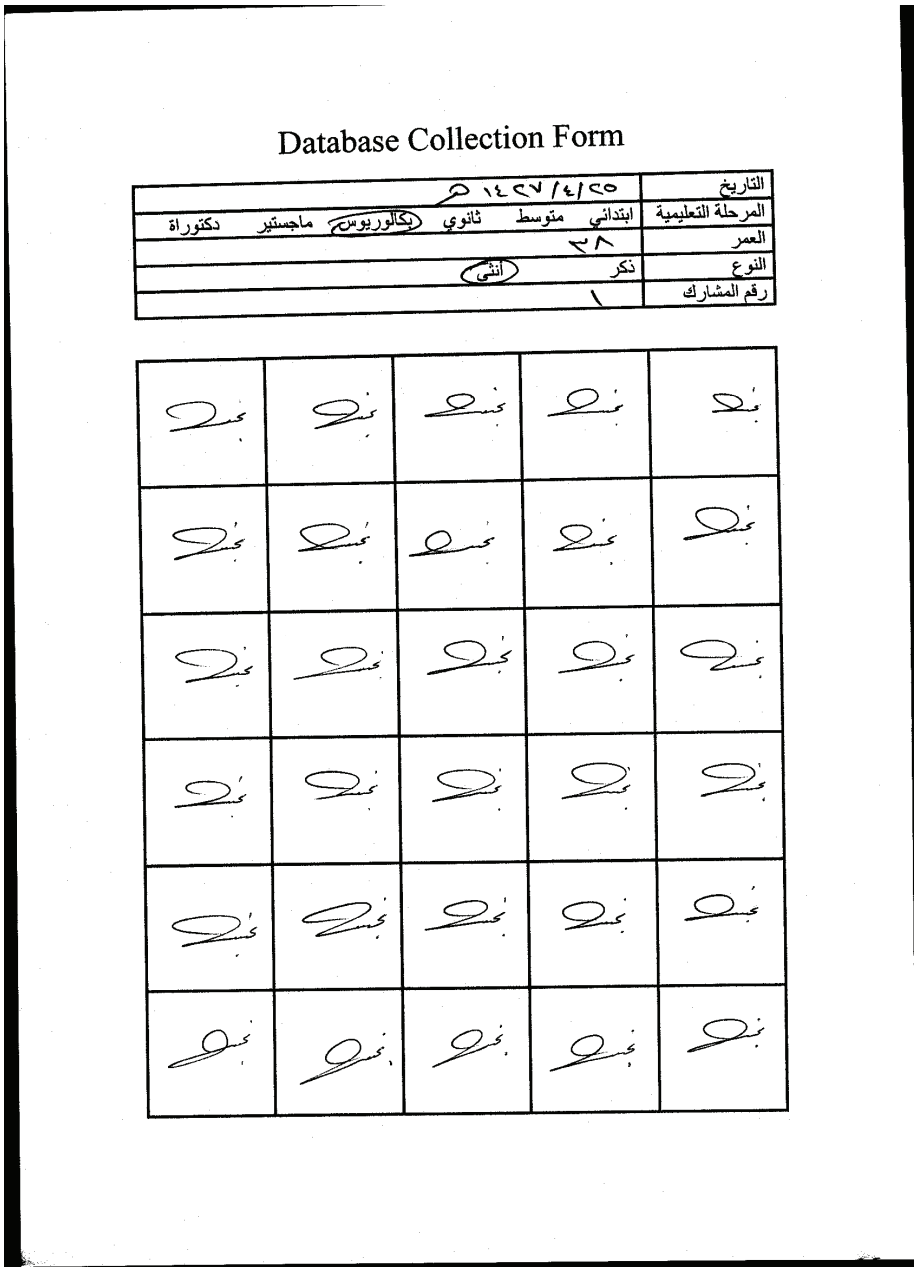

Figure 4. Skewed database form image during scanning. It requires estimation of the skew angle to calibrate prototype separation points.

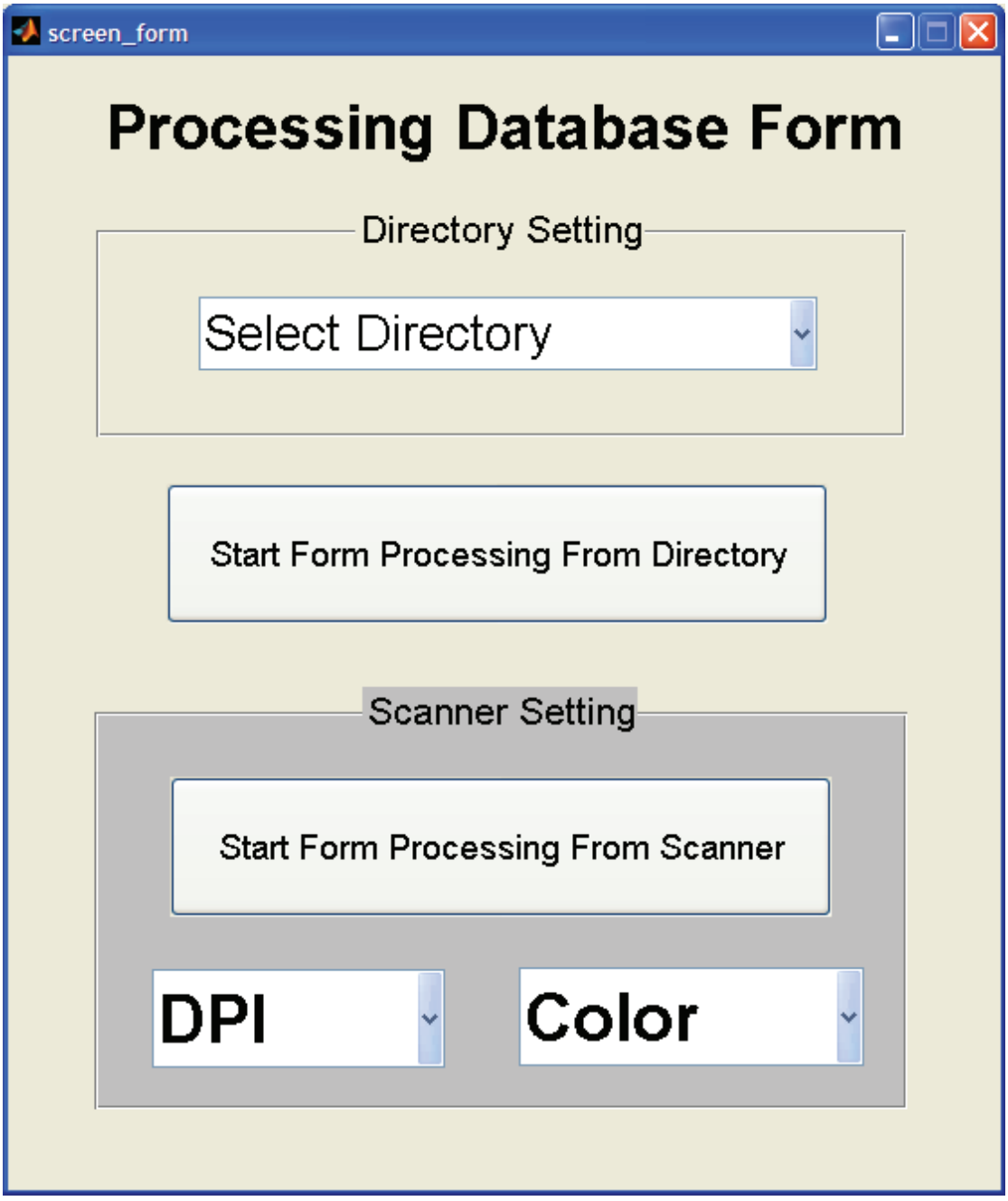

Figure 5. The GUI for the database processing tool.

REFERENCES

AbdelRaouf, A., Higgins, C. A., and Khalil, M. 2008. A Database for Arabic Printed Character Recognition. Image Analysis and Recognition, ICIAR 2008, LNCS 5112, pp. 567–578.

Al-Ma'adeed, S., Elliman, D., and Higgins, C. A. 2004. A Data Base for Arabic Handwritten Text Recognition Research. The International Arab Journal of Information Technology, Vo.1, No.1.

Al-Ohali, Y., Cheriet, M., and Suen, C. Y. 2003. Databases for recognition of handwritten Arabic cheques. Pattern Recognition, 36(1):111–121.

Al Shalabi, H. M., Hasan, M. F., and Ali, A. M. 2005. A New Data Base Scheme Arabic Handwriting Recognition by Hopfield Neural Networks Algorithm. Journal of Computer Science 1 (2): 204-206.

El Abed, H., and Margner, V. 2007. The IFN/ENIT-database - a tool to develop Arabic handwriting recognition systems, The 9th International Symposium on Signal Processing and Its Applications (ISSPA 2007), pp.1-4, 12-15, Feb. 2007.

Gonzalez, R. C., Woods, R. E., and Eddins, S. L. 2003. Digital Image Processing Using MATLAB. Pearson Prentice Hall, Upper Saddle River, New Jersey, USA.

Jain, R., Kasturi, R., and Shunck, B. 1995. Machine Vision. McGraw-Hill and China Machine Press, Beijing, China.

Kharma, N., Ahmed, M., and Ward, R. 1999. A new comprehensive database of handwritten Arabic words, numbers, and signatures used for OCR testing, IEEE Canadian Conference on Electrical and Computer Engineering, vol. 2, pp. 766-768.

Lorigo, L. M., and Govindaraju, V. 2006. Offline Arabic Handwriting Recognition: A Survey. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 28, No. 5.

Nixon, M. S., and Aguado, A. S. 2008. Feature Extraction and Image Processing. Academic Press, Oxford, UK.

Otsu, N. 1979. A threshold selection method from gray-level histograms. IEEE Transactions on Systems, Man, and Cybernetics 9: 62-66.

Pechwitz, M., Maddouri, S. S., Märgner, V., Ellouze, N., and Amiri, H. 2002. IFN/ENIT-Database of Handwritten Arabic Words, The 7th Colloque International Francophone sur l'Ecrit et le Document , CIFED 2002, Oct. 21-23, Hammamet, Tunis.

Pechwitz, M., and Maergner, V. 2003. HMM based approach for handwritten Arabic word recognition using the IFN/ENIT - database, Proceedings of the Seventh International Conference on Document Analysis and Recognition (ICDAR'03), pp. 890-894, 3-6.

Saabni, R., and El-Sana, J. 2009. Efficient Generation of Comprehensive Database for Online Arabic Script Recognition. ICDAR, 1231-1235.

Slimane, F., Ingold, R., Kanoun, S., Alimi, A., and Hennebert, J. 2009. A New Arabic Printed Text Image Database and Evaluation Protocols. International Conference on Document Analysis and Recognition (ICDAR 09), July 26 - 29, Barcelona, Spain.

Smith, T. S. 2006. MATLAB Advanced GUI Development, Dog Ear Publishing, Indianapolis, IN, USA.

Snyder, W. E. and Qi, H. 2004. Machine Vision. Cambridge University Press, Cambridge, England.

Received 7/5/1431; 21/4/2010, accepted 22/6/1431; 5/6/2010