International Journal of Computer Science & Applications
Vol. II, No. II, pp. 131 - 144

Preprocessing and Image Enhancement Algorithms for a Form-based Intelligent Character Recognition System

Dipti Deodhare, NNR Ranga Suri
Centre for AI and Robotics,
Raj Bhavan Circle,
Bangalore, India.
Email:{dipti, nnrsuri}@cair.res.in.

R. Amit
Computer Science Dept.,
Univ. of Southern California,
USA.
amitr@usc.edu.

Abstract

A Form-based Intelligent Character Recognition (ICR) System for handwritten forms, besides others, includes functional components for form registration, character image extraction and character image classification. Needless to say, the classifier is a very important component of the ICR system. Automatic recognition and classification of handwritten character images is a complex task. Neural Networks based classifiers are now available. These are fairly accurate and demonstrate a significant degree of generalisation. However any such classifier is highly sensitive to the quality of the character images given as input. Therefore it is essential that the preprocessing components of the system, form registration and character image extraction, are well designed. In this paper we discuss the form image registration technique and the image masking and image improvement techniques implemented in our system as part of the character image extraction process. These simple yet effective techniques help in preparing the input character image for the neural networks-based classifiers and go a long way in improving overall system accuracy. Although these algorithms have been discussed with reference to our ICR system they are generic in their applicability and may find use in other scenarios as well.
Keywords: Form-based ICR, skew correction, form masking, character image extraction, neural networks.

1 Introduction

Manual data entry from hand-printed forms is very time consuming - more so in offices that have to deal with very high volumes of application forms (running into several thousands). A form-based Intelligent Character Recognition (ICR) System has the potential of improving efficiency in these offices using state-of-the-art technology. An ICR system typically consists of several sequential tasks or functional components, viz. form designing, form distribution, form registration, field-image extraction, feature-extraction from the field-image, field-recognition (here by field we mean the handwritten entries in the form). At the Centre for Artificial Intelligence and Robotics (CAIR), systematic design and development of methods for the various sub-tasks has culminated into a complete software for ICR. The CAIR ICR system uses the NIST (National Institute for Standards and Technology, USA) neural networks for recognition [4, 5, 6]. For all the other tasks such as form designing, form registration, field-image extraction etc. algorithms have been specially designed and implemented. The NIST neural networks have been trained on NIST's Special Database 19 [8, 3, 7]. The classification performance is good provided the field, i.e. the handwritten character entry in the form, is accurately extracted and appropriately presented to the neural network classifiers. Good preprocessing techniques preceding the classification process can greatly enhance recognition accuracy [15, 12]. In what follows, in Section 2, we discuss in some detail a robust and efficient algorithm for form image registration. Thereafter in Sections 3, 4 and 5, three techniques for image masking and image improvement are discussed and their influence on overall system performance is illustrated with several examples.

2 Scanned Form Image Registration

Forms are filled and mailed from all over. As a result they are received folded and are often dog-earred and smudged. Moreover, use of stapling pins, paper clips etc. introduces a lot of noise in the form image. Due to this and given that a large number of forms have to be processed in a short time, the form registration algorithm needs to be robust to noise and highly efficient. The form registration also has to be very accurate since the accuracy of the field image extraction and hence field recognition depends on it. Since the form is required to be designed using our ICR interface,

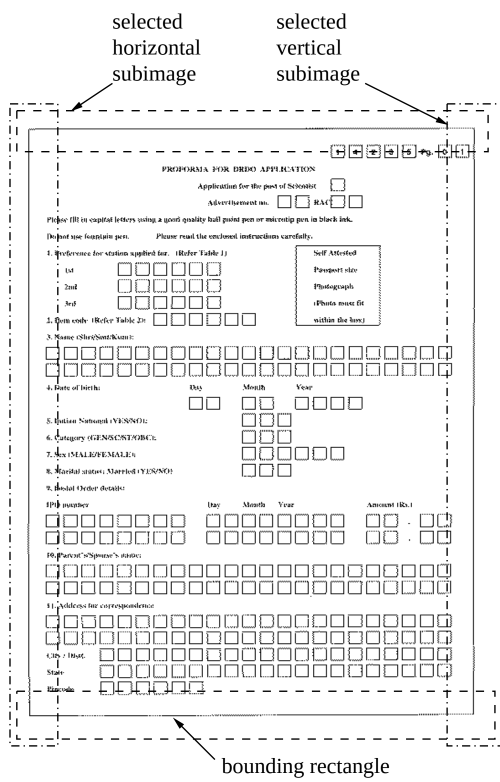


Figure 1: Form template with marked sub-images

the form layout and hence the field positions are already known. Therefore field-image extraction is a straight forward process *provided* the form is positioned accurately on the scan-bed during scanning. This is unlikely and skew and shift are always introduced in the form image during the scanning process. Form registration is the process by which the skew angle and the shift in the form image are assessed and corrected for.

Several methods have been developed for skew angle detection. Hough transform-based methods described in [1, 10, 11, 13, 14] are computationally expensive and depend on the character extraction process. A method based on nearest-neighbor connection was proposed in [9], but connections with noise can reduce the accuracy of that method. Although a method was proposed in [17] which deals with gray-scale and color images as well as binary images, the accuracy of that method depends upon the way the base line of a text line was drawn. In [2], Chen and Lee present an algorithm for form structure extraction. Although the algorithm can tolerate a skew of around 5° [2], as such it is not used for skew correction. This basic algorithm has been improvised here and used in conjunction with a bounding rectangle introduced in the form during the form design process, to arrive at a method that is efficient and highly robust to noise for skew correction.

To assist robust form registration, a bounding rectangle of user defined width and height (referred to as RECT_WIDTH and RECT_HEIGHT respectively in the following discussion) is introduced in the form during form design. All fields of the form are constrained to be contained within this rectangle as shown in Figure 1. A descriptive version of the proposed registration algorithm is presented here. For the detailed version, refer to the algorithm described in [16].

Algorithm : Form Registration
Input: Scanned form image. *Output:* Form image after skew and shift correction.
begin

1. Extract adequate portions from the top and bottom half of the form image for detecting the two horizontal sides of the rectangle. Similarly for detecting the vertical sides, extract sub-images from the left and right halves of the form image, as shown in Figure 1.
2. Divide the sub-images into strips and project each strip. Using the projection values along the scan lines detect the line segments in each strip and then the corresponding start points.
3. Use the line tracing algorithm similar to that in [2] with a 3×3 window for connectivity checking. Having obtained segment points using line tracing, fit a line to these points using the pseudo-inverse method to obtain the slope.
4. Starting with the middle strip in each sub-image, merge the line segments with the line segments in the strips on either directions of the middle strip to obtain full length lines. This results in four sets of lines corresponding to the four sub-images called the top_line_set, the bottom_line_set, the left_line_set and the right_line_set.
5. Identify a pair of lines, one from the top_line_set and the other from the bottom_line_set as the top edge and the bottom edge of the rectangle respectively, if they are almost parallel to each other and the perpendicular distance between them is equal to the height of the rectangle. Similarly identify the left and right edges of the bounding rectangle using the left_line_set and the right_line_set.
6. To improve the estimates, discard the outliers from the coordinates array of points of the detected edges. Fit a line using least squares for points in the new array and return this array along with the estimated slope and offset. Use the slope values of the four lines to assess the skew in the form and rotate the form for correcting this skew. Perform the same transformation on the edge points and recompute the slope and offset values of the edges after rotation. Use these new offset values for shift correction.

end

The above algorithm has been tried out on form images with different skew angles. The forms were scanned by a HP ScanJet 6350C with a resolution of 300dpi. Figure 2 summarizes the different stages involved in the process of detecting the bounding rectangle edges for form

(a)
(b)
(c)
(d)
(e)

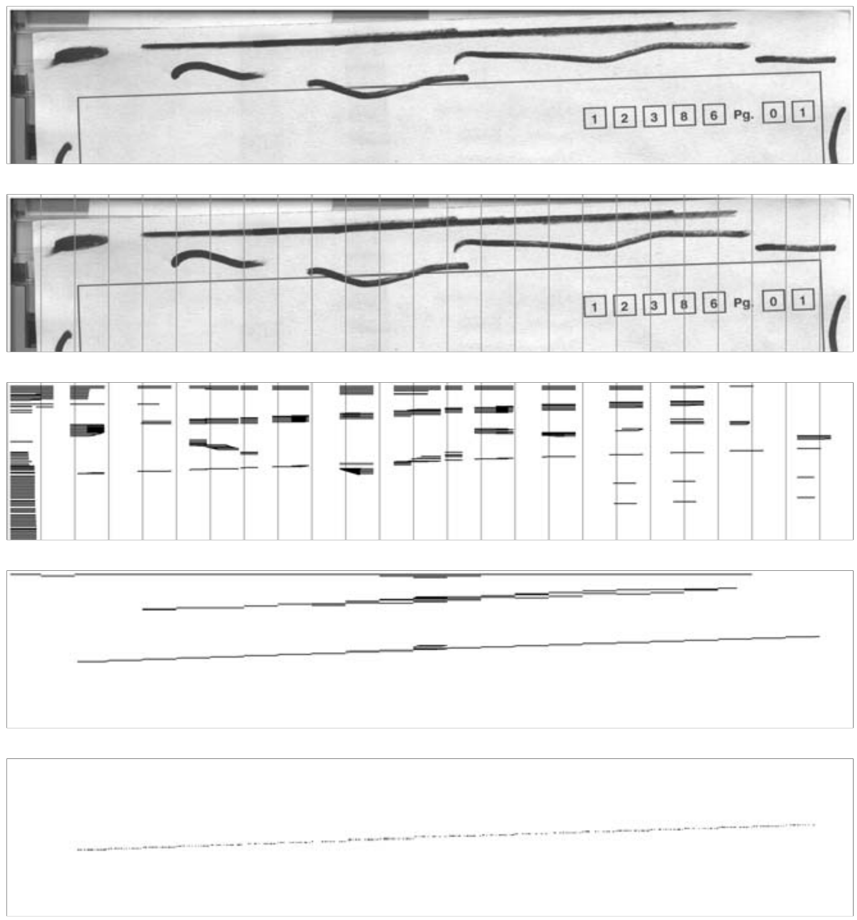
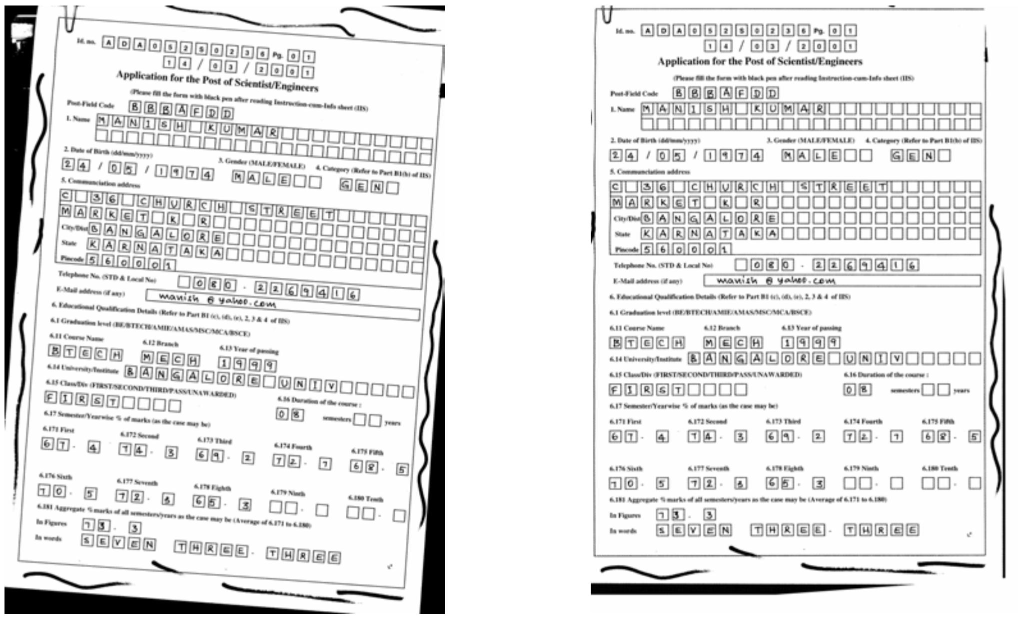


Figure 2: (a) Top horizontal sub-image. (b) Sub-image division into vertical strips. (c) Detected segments. (d) Lines obtained after merging the segments. (e) The top edge of the bounding rectangle.

Actual Skew	Measured Skew
$2°$	$2.041°$
$5°$	$4.982°$
$7°$	$6.996°$
$10°$	$9.943°$
$13°$	$12.943°$
$18°$	$17.94°$

Table 1: Actual skew vs skew measured by the registration algorithm



(a) (b)

Figure 3: (a)Input scanned form image, (b) Form image after skew and shift correction

image registration. A sample output for one such form has been shown in Figure 3. Since the algorithm operates only on the sub-images extracted from the margins of the form image, a lot of artificial noise was introduced along the margins, for testing purposes. Listed below in Table 1 are results from some of the test cases.

The number of strips that the sub-images are divided into influences the performance of the algorithm both in terms of accuracy and efficiency. Hence the number of strips should be decided based on scanned image dimensions. The typical values used in the implementation are: number of strips along the width = 25 and number of strips along the height= 35.

3 Local Registration and Field Box Masking

In Section 2, a robust and efficient algorithm for registration of scanned images was presented. The form registration algorithm performs a global registration of the form and though an essential ingredient of the processing scheme is not sufficient. Form printing and subsequent form scanning for ICR introduces non-uniform distortions in the form image. This necessitates a local registration of field boxes after a global registration of the form image has been performed. This is because the skew and shift parameters computed by the registration algorithm for the current scanned form are used to correct the box position values stored by the system during the form design process. However these corrected boxes need not exactly coincide with the boxes in the registered image of the scanned form. In Figure 4, the light gray boxes represent the box positions corrected for the computed skew and shift and dark gray boxes represent the actual box positions in the registered image. Moreover the field boxes themselves may undergo a structural change due to the process of printing and scanning further complicating the process of field box masking. Accurate field box masking is essential for character image extraction because if portions of the box remain in the final character image presented to the neural network classifier the results are often inaccurate. To circumvent these problems the following correlation based algorithm has been devised to perform local registration and masking of the field boxes.

Algorithm: Mask form image field boxes.
Input: Form image, list of ideal registration points for field boxes (as recorded by the form design module.)
Output: Masked image of the form, list of registration points corrected for the current form.
Begin

1. Initialize corrected registration points list as an empty list.
2. For every field box perform steps 3 to 8.
3. Set Top_Left (x, y) to top left corner coordinate of current field box from the list of ideal registration points.
4. Set Bottom_right (x, y) to bottom right corner coordinate of current field box from the list of ideal registration points.
5. In the neighborhood of the field box ideal position, locate the position with maximum correlation in terms of the number of matching ink points. (The neighborhood is defined by an N x N grid centered at the ideal field box position.) Label this box as Max_corr_box and its corner points as Max_corr_top_left and Max_corr_bottom_right respectively, and the correlation value as Max_corr.
6. Stretch each side of the Max_corr_box, one at a time, to a maximum distance of DELTA and store the resulting corner coordinates each time the correlation exceeds THRESHOLD (= THRESH_PERCENT * Max_corr).
7. Draw on the form image, in background colour, the Max_corr_box, and each box whose coordinates have been stored in step 6.
8. Append Max_corr_box corner coordinates to corrected registration points list.
9. Return the masked image and the corrected registration points list.

End

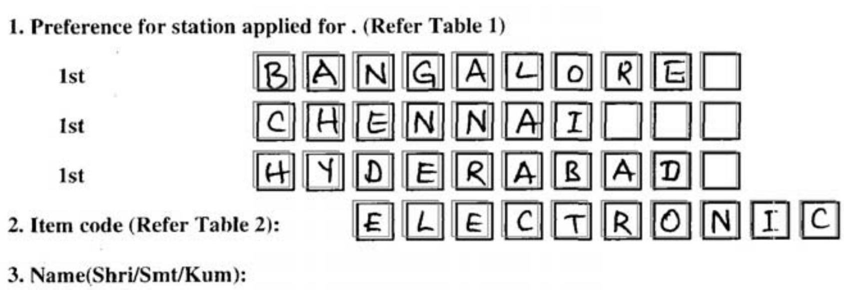
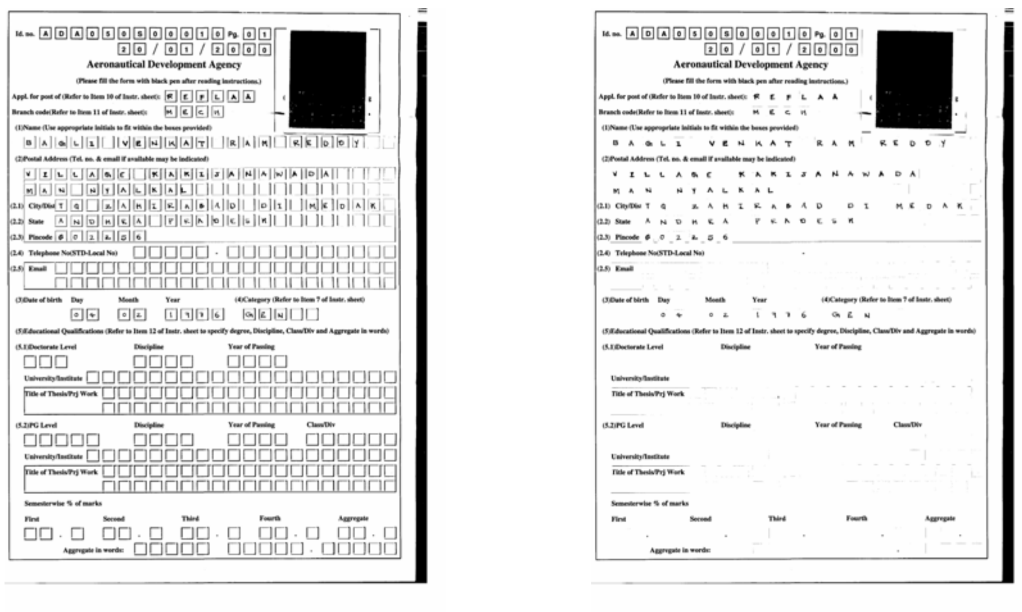


Figure 4: Ideal and actual field box positions

Figure 5 demonstrates the effectiveness of the above described local registration and correlation-based field box masking algorithm. The image in Figure 5(a) is the image obtained after attempting a masking of the field boxes immediately after applying the registration algorithm of Section 2. The result is very poor since the sides of several boxes remain. When the masking is performed in conjunction with the local registration algorithm described above the results are visibly superior as seen in Figure 5(b).



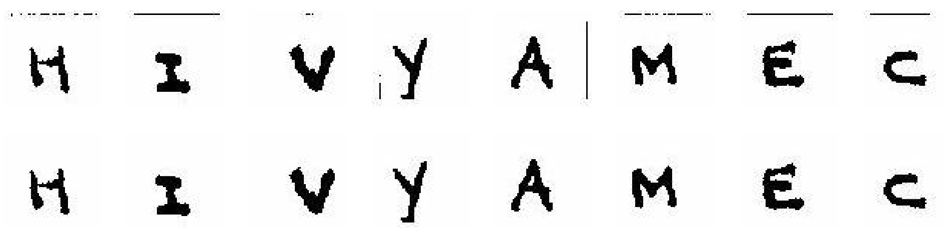
(a) (b)

Figure 5: (a) Field Box Masking without Local registration (b) Field Box Masking with Local Registration

4 Noise Cleaning along Character Image Boundaries

As demonstrated by the example in Figure 5, form registration followed by local registration of the field boxes enables a very good fix on the field box position. Consequently, most of the time the masking process cleans up the field box accurately. However, often, fine residual lines remain leading to wrong classification of the character image. Some such character images have been shown in Figure 6. Consider for example the first character of the top row. This is the character image of the letter ‘H’ obtained after field box masking. When this image is presented to the neural network for classification it is classified as the letter ‘W’. The fine line left unmasked at the top of the image confuses the neural network completely. To overcome this problem, an image projection-based algorithm has been implemented. The first character image in the bottom row of Figure 6 is the character image obtained after applying this algorithm to the image of character ‘H’ in the top row of Figure 6. The neural network correctly classifies this image as the character ‘H’. Table 2 lists the classification results of the character images of Figure 6 before and after applying the algorithm for cleaning noise along the boundaries. As mentioned in the introduction the classifier is the NIST neural network for uppercase letters.

Algorithm: Projection based algorithm for cleaning noise along the boundaries of the character image
Input: Character image obtained after applying the masking algorithm of Section 3.
Output: Character image with noise removed from its boundaries.



(1) (2) (3) (4) (5) (6) (7) (8)

Figure 6: Top Row: Character image after field box masking. Bottom Row: Character image obtained after the noise along the image boundaries in the masked image is cleaned.

Sr.No.	*Result before boundary cleaning* (Top row)	*Result after boundary cleaning* (Bottom row)
1.	'W'	'H'
2.	'W'	'I'
3.	'W'	'V'
4.	'X'	'Y'
5.	'N'	'A'
6.	'W'	'M'
7.	'W'	'E'
8.	'U'	'C'

Table 2: Classification results of the neural network on the character images in Figure 6

Begin

1. Project the image vertically and record the projection values.
2. Similarly project the image horizontally and record the projection values.
3. To clear the noise along the left boundary of the image do steps (a) to (c) given below.
   (a) Using vertical projection values, identify the left most column c with a non-zero projection value.
   (b) Starting with such a column and going up to 1/8 the width of the image from the left, find out the column c' which is to the right side of c and whose projection value is less than some preset threshold. (In our implementation this value has been set to 3.) This condition locates the gap between the boundary and the handwritten character. Column c' will be the rightmost column to the right of which the ink points corresponding to the handwritten character image will be found.
   (c) If the distance between c and c' is less than the possible character width, set the pixel values between the columns c and c' to the background value. This condition takes care of the situation where the masking is perfect and no residual noise lines are left along the image boundaries.
4. To clear the noise along the right boundary of the image do steps (a) to (c) given below.
   (a) Using vertical projection values, identify the right most column c with a non-zero projection value.
   (b) Starting with such a column and going up to 1/8 width of the image from right, find out the column c' which is to the left side of c and whose projection value is less than some preset threshold value.

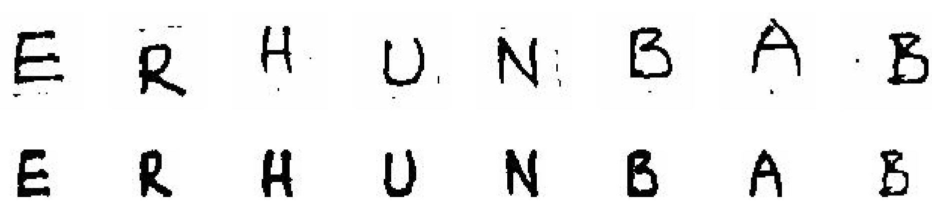
(c) As in step 4(c) above, set the pixel values between the columns c and c' to the background value if the distance between c and c' is less than the possible character width.

5. Repeat steps 3 and 4 and using horizontal projection values to clear the noise along the top and the bottom boundaries of the image.

End

5 Omitting Isolated Noise in Character Images

Once the character image is extracted from the form image, it is normalized before it is given as input to a neural network for recognition. This is done to reduce the variability in the input data that the neural network has to deal with - a mechanism usually employed to keep generalization demands, on the network, moderate. This results in smaller networks requiring less training time for convergence. Since we are using NIST designed neural networks we need to conform to the format of image input that the NIST neural networks expect. As a result the character image is normalized to fit tightly within a 20 x 32 pixel region and then centered in a pixel image of size 32 x 32. Before the image can be normalized to fit tightly in a 20 x 32 pixel region the exact bounding box within which the handwritten character lies has to be determined. Isolated noise blobs lead to inaccurate detection of the bounding box. This in turn leads to inaccurate recognition. For example refer to Figure 7. The character images in the top row of this figure are those obtained after applying the boundary noise cleaning algorithm of Section 4. Each image in the top row has tiny specks and/or fine noise lines. These may be introduced on the image either by the printing process, the mailing process or due to dust particles present on the ADF (Automatic Document Feeder) or on the scanbed of the scanner. If these noise blobs are not appropriately ignored or omitted during the bounding box detection process the results can be inaccurate. Table 3 lists the classification results of the images in Figure 7. When a naive bounding box detection technique is employed on the images in the top row, the results are inaccurate. When the neighborhood based method, discussed below, is used the results are accurate. The bottom row of Figure 7 shows the normalized image of the character images in the top row of the same figure. It is evident that the bounding box detection process has ignored the noise blobs as desired. The algorithm devised to obtain the correct bounding box boundaries is described next.



(1) (2) (3) (4) (5) (6) (7) (8)

Figure 7: Top Row: Character image obtained after the noise along the image boundaries in the masked image is cleaned. Bottom Row: Normalized images of the character images in the top row.

Algorithm: Neighborhood search-based method to omit isolated noise blobs in the character image while computing the image bounding box
Input: Character Image obtained after applying the boundary noise removal algorithm of Section 4.
Output: Coordinates of the top left and the bottom right corners of the bounding box of the input character image.

Sr.No.	*Result without noise blobs omission method* (Top row)	*Result with noise blobs omission method* (Bottom row)
1.	'N'	'E'
2.	'X'	'R'
3.	'Y'	'H'
4.	'S'	'U'
5.	'M'	'N'
6.	'I'	'B'
7.	'P'	'A'
8.	'A'	'B'

Table 3: Classification results of the neural network on the character images in Figure 7

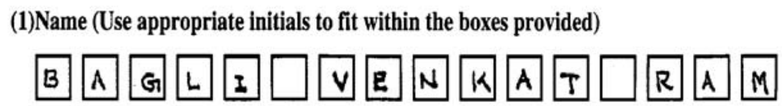
Begin

1. Set Boundary_Top_Left(X,Y) equal to (Char_Image_Width, Char_Image_Height).
2. Set Boundary_Bottom_Right(X,Y) equal to (0,0).
3. Starting from (0, 0) do steps 4 to 14 for all points in the image.
4. Set Curr_Point as the next point in the image. (The very first point is (0, 0)). If all points are exhausted then end.
5. If Curr_Point is an ink point
   Take Curr_Point as the center point of an N_HOOD x N_HOOD grid.
   Set COUNT = number of ink points in this grid.
   Else
   Go to step 4.
6. If COUNT $<=$ (DENSITY * N_HOOD * N_HOOD)
   Curr_Point is an isolated noise point. Go to step 4.
7. Set Left = left most ink point in the N_HOOD x N_HOOD grid centered at Curr_Point.
8. If Boundary_Top_Left.X $>$ Left, set Boundary_Top_Left.X = Left
9. Set Top = top most ink point in the N_HOOD x N_HOOD grid centered at Curr_Point.
10. If Boundary_Top_Left.Y $>$ Top, set Boundary_Top_Left.Y = Top
11. Set Right = right most ink point in the N_HOOD x N_HOOD grid centered at Curr_Point.
12. If Boundary_Bottom_Right.X $<$ Right then set Boundary_Bottom_Right.X = Right
13. Set Bottom = bottom most ink point in the N_HOOD x N_HOOD grid centered at $Curr\_Point$.
14. If Boundary_Bottom_Right.Y $<$ Bottom, then set Boundary_Bottom_Right.Y = Bottom

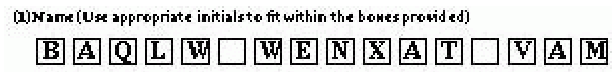
End

6 Results Summary and Conclusion

Our ICR system has been successfully deployed for recruitment in an Indian government office. Approximately 700 forms were processed. The form designed had three pages. All examples in this paper have been taken from filled application forms received in the above mentioned recruitment exercise. Our ICR system proved to be efficient and reduced the time required for processing

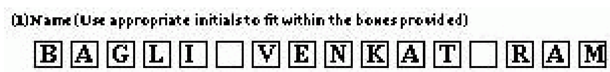


(a)



(b)

(c)



(d)

Figure 8: (a) An extract from a scanned form image. (b) Recognition output after masking, (c) Recognition output after masking and noise cleaning along boundaries, (d) Recognition output after masking, noise cleaning along boundaries and neighborhood search-based bounding box detection.

the applications considerably. Figure 8 summarizes the effect of the techniques discussed above on the final accuracy of the system.

The above exercise guided the upgrade of the software and after fine tuning some of the user interface utilities, our ICR system was again benchmarked on sample SARAL forms made available to us by the Income Tax Office, at Infantry Road, Bangalore. 150 sample SARAL (Form 2D) forms, used for filing the financial returns of an employed individual were filled in Range-13 of Salary Circle and processed using the ICR system. The 3-page SARAL form is shown in Figure 9. The results of the exercise have been tabulated in Table 4 below. A note regarding the entry corresponding to “Dictionary” in the table. In a typical form there are several fields that can take values only from a predefined set, for example the SEX field in a form can take only two values - MALE/FEMALE. The system allows the user to create a “dictionary” corresponding to such fields. For these fields, after performing the character recognition in individual boxes, all the characters corresponding to the field are concatenated into a single string. The distance of this string, measured in terms of a string metric known as the *Levenstein metric*, from strings in the dictionary associated with this field is calculated. The string is replaced by the dictionary string closest to this string. This dramatically improves the accuracy of the system output.

The system was also successfully deployed at the Centre for AI and Robotics (CAIR), India. Two all India recruitments were undertaken - one in 2003 and the other in 2004. Some details are included in Table 5 to give an indication of the volume of forms handled by the software. System accuracy for the recruitment done through the system in 2003 are included in Table 6.

Field Type	*No. of Mis-classifications*	*Classification Accuracy*
Dictionary	3(total = 585)	99.49%
Numeric	157(total = 2552)	93.85%
Upper case	249(total = 2885)	91.37%

Table 4: Classification results for the Income Tax SARAL forms

Year	*No. of Forms Distributed*	*No. of Forms Processed*	*Total no. of Post Categories*	*Total Posts*
2003	5000	3272	4	32
2004	3900	2916	4	17

Table 5: Details of recruitments conducted at CAIR using the system.

To conclude, a robust algorithm has been described for measuring and correcting the skew and shift values that are present in a scanned form image. Subsequently, three techniques, viz. (*i*) field box masking, (*ii*) noise cleaning along character image boundaries and (*iii*) neighborhood search-based bounding box detection, that together comprise the handwritten character extraction process have been presented. The necessity and impact of these methods on the overall performance of the ICR system has been systematically illustrated by examples. The effectiveness of these algorithms has been convincingly proved by the fact that the system performed with adequate accuracy in real life recruitment exercises requiring the processing of handwritten application forms.

Acknowledgments The authors would like to thank Director CAIR for the support and encouragement received by them for the work reported in this paper. The authors would also like to extend their thanks to Mr. D. S. Benupani, Additional Commissioner of Income Tax, Bangalore for providing the SARAL form benchmark data.

References

[1] B. B. Chaudhuri and U. Pal. A complete printed bangla ocr system. *Pattern Recognition*, 31(5):531–549, May 1998.

[2] J. L. Chen and H. J. Lee. An efficient algorithm for form structure extraction using strip projection. *Pattern Recognition*, 31(9):1353–1368, May 1998.

[3] M. D. Garris, J. L. Blue, G. T. Candela, D. L. Dimmick, J. Geist, P. J. Grother, S. A. Janet, and C. L. Wilson. Nist form-based handprint recognition system. *NIST Internal Report 5469 and CD-ROM*, July 1994.

[4] M. D. Garris, C. L. Wilson, and J. L. Blue. Neural network-based systems for handprint ocr applications. *IEEE Transactions on Image Processing*, 7(8):1097–1110, August 1998.

Field Type	*No. of Mis-classifications*	*Classification Accuracy*
Dictionary	43(total = 7398)	99.42%
Numeric	1811(total = 28567)	93.77%
Upper case	3873(total = 33257)	88.35%

Table 6: Classification results for the CAIR recruitment, 2003

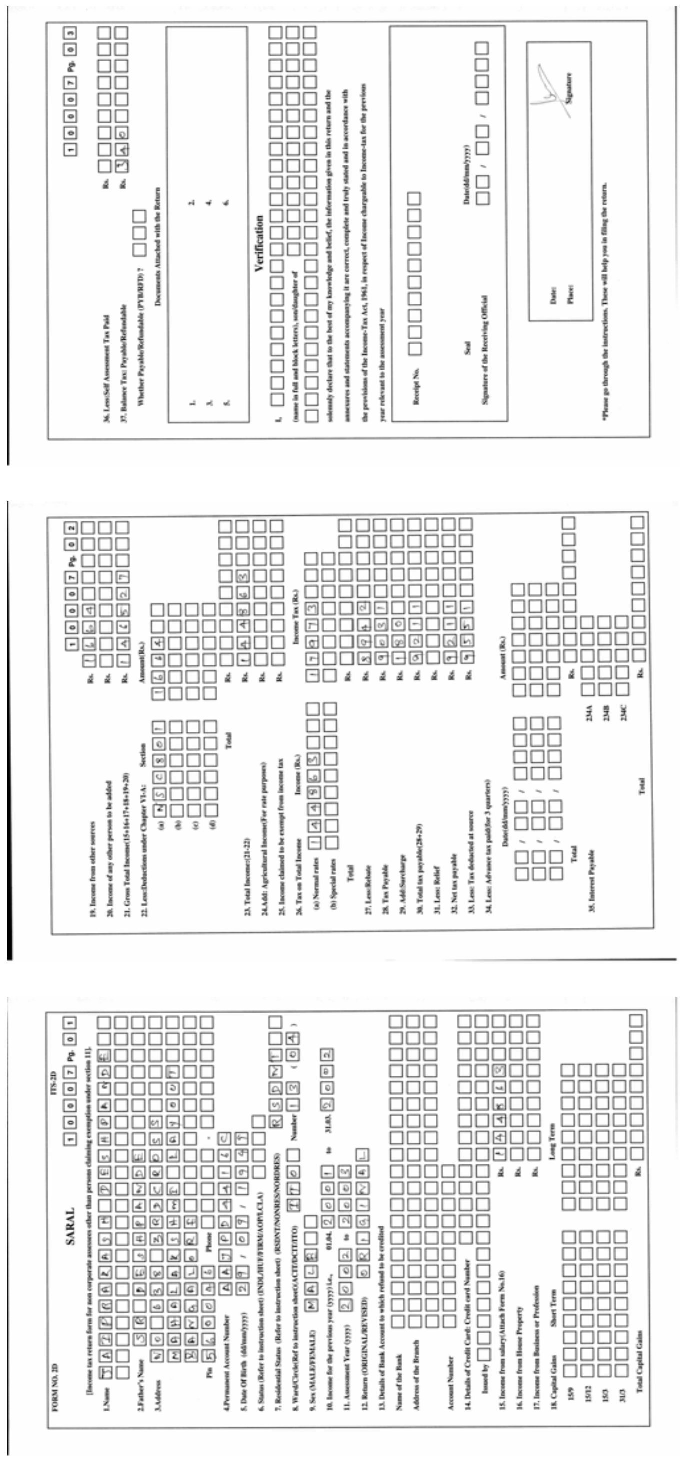


Figure 9: A filled sample of 3-page SARAL form

[5] P. J. Grother. Karhunen loève feature extraction for neural handwritten character recognition. *Applications of Artificial Neural Networks III, SPIE, Orlando*, 1709:155–166, April 1992.

[6] P. J. Grother. Karhunen loève feature extraction for neural handwritten character recognition. *NIST Internal Report 4824*, April 1992.

[7] P. J. Grother. Handprinted forms and characters database, nist special database 19. *NIST Technical Report and CD-ROM*, March 1995.

[8] Patrick J. Grother. Nist special database 19 handprinted forms and characters database. Technical report, NIST, March 1995.

[9] A. Hashizume, P. S. Yeh, and A. Rosenfeld. A method of detecting the orientation of aligned components. *Pattern Recognition Letters*, 4:125–132, 1986.

[10] S. C. Hinds, J. L. Fisher, and D. P. D'Amato. A document skew detection method using run-length encoding and the hough transform. *Proc. 10th Int. Conf. Patt. Recogn. (ICPR) (Atlantic City, NJ)*, pages 464–468, June 1990.

[11] D. S. Le, G. R. Thoma, and H. Wechsler. Automated page orientation and skew angle detection for binary document images. *Pattern Recognition*, 27(10):1325–1344, 1994.

[12] G. Nagy. Twenty years of document image analysis in pami. *IEEE Trans. Pattern Anal. and Mach. Intell.*, 22(1):38–62, January 2000.

[13] Y. Nakano, Y. Shima, H. Fujisawa, J. Higashino, and M. Fujinawa. An algorithm for the skew normalization of document image. *Proc. 10th Int. Conf. Patt. Recogn. (ICPR) (Atlantic City, NJ)*, pages 8–13, June 1990.

[14] L. O'Gorman. The document spectrum for page layout analysis. *IEEE Trans. Pattern Anal. and Mach. Intell.*, 15(11):1162–1173, November 1993.

[15] Z. Shi and V. Govindraju. Character image enhancement by selective region-growing. *Pattern Recognition Letters*, 17:523–527, 1996.

[16] N. N. R. Ranga Suri, Dipti Deodhare, and R. Amit. A robust algorithm for skew and shift correction for a handprinted form-based icr system. *Proc. 4th Int. Conf. Advances in Pattern Recognition and Digital Techniques (Calcutta)*, pages 411–417, December 1999.

[17] H. Yan. Skew correction of document images using interlace cross-correlation. *CVGIP: Graphical Models Image Process.*, 55(6):538–543, November 1993.