Automatic Inspection of Braille Character:A Vision based Approach.

[1a] J. Bhattacharya , [2a] S.Majumder and [3b] G.Sanyal

[a]*Central Mechanical Engineering Research Institute*
Durgapur 713209,West Bengal,India
[b]*National Institute of Technology*
Durgapur 713209,West Bengal,India

***Abstract*—Analysis of the printing quality of a Braille printer is an essential task as any minute nonuniformity in the braille dot spacing will lead to the misinterpretation of the whole document.The paper uses a multi-parameter based feature detection algorithm which efficiently detects the embossed dots from a braille print.This is further utilized to make a vision based analysis of the achieved printing quality of the braille printer developed by CMERI.Performance analysis of the proposed method with other conventional approaches so far used for this purpose clearly shows that the multi-parameter based feature detection method gives a higher accuracy even for a degraded image.A simple approach is also proposed for the recognition of the interpoint braille characters from a single side and generation of English text for both sides of the print simultaneously.This can be utilized for regeneration of the braille print from the hard copy of the document.**

***Index Terms*—Braille Character Recognition, computer vision,Generalized Feature Vector, feature detection,interpoint braille.**

I. Introduction

Braille system originally devised by Louis Braille provides an unique means using which visibly challenged persons carry out reading and writing. Essentially in Braille any alphabet is represented by a cell containing six or eight convex dots.A typical example of which is shown in figure 1 using Bharti Braille(the Braille scheme adopted in India) which applies six dot cell. The geometrical dimensions specifying these dots are shown in figure 2.Although a few manufacturers globally manufacture and supply Braille print but are often beyond financial reach of the common (Indian) visually challenged persons.This has inspired us to design and develop a Braille printer at an affordable cost. A major problem however faced by the designers is how to measure and ensure the quality of braille printout which complies the braille specification of dot geometry.Normally a blind person is used to check the quality of the braille printed material.In general braille printout of a machine is put to a qualified reader with the knowledge of braille.He or she then assess the quality by reading the text using his or her finger.This by far is qualitative and often the assessment of the quality is subject to the reader's knowledge and experience in reading various printout produced by any specific printer.As such the lack of well defined standard measuring technique to ensure the dot quality led us to develop a computer vision based approach.This paper therefore presents a quantitative method to evaluate braille printing quality and removes the scope of qualitative and subjective interpretation based on ones knowledge skill and acquintances.In this approach a multiparameter based feature detector is used to extract the braille dots from the braille printing and measurement of the dot spacings were taken for examining the quality of printing.This approach provides the necessary means to assess the quality of Braille printing in a quantitative manner and removes the qualitative assessment procedure used for a long time.

This paper is organized in the following manner.Section I discusses the basic background of the problem. Section II provides the analysis of the printing quality using a vision based approach whereas section III suggests a technique for braille character recognition. Finally, discussion and conclusion is presented in section IV.

II. Vision based printing analysis

Braille prints are mainly assessed by their dot spacings.Uniform spacings are very essential for a braille print as the entire code may be wrongly interpreted with a slight misplacement of a dot.Inorder to calculate the inter-dot spacings the dots are required to be detected.This is achieved using a digitized image of the braille print.It is obvious that detecting the dots from a similar colored background is a very challenging task.A number of different techniques have been tried to obtain a good quality digitized image of a braille print .In the past, researchers have tried to apply common image processing techniques like canny edge detection [1]and image thresholding with beta distribution [2] for dot detection.The performance of these procedures depends on a large extent on the digital image quality.It may yield reasonable results on a

Fig. 1. A test print using the braille printer developed by CMERI.

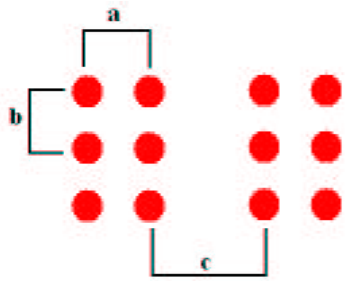


Fig. 2. Figure shows the dot spacing used for the braille print.The intradot horizontal and vertical spacing a and b is 2.4 and 2.1 respectively.The horizontal spacing between characters c is3.5.

very high quality image but fail to provide desired output for a slightly degraded quality image as shown in figure3b.It is thus obvious that a single parameter is not sufficient to detect the dots with reasonable consistency.Figure 4 and 5 shows the result of applying classical edge detection and thresholding operations on sample data shown in figure 3b .It clearly depicts that both canny and multi-level thresholding procedures fail to recognize the dot positions.In case of application of canny, the edges are detected as disjoint dots and it is absolutely impossible to extract any dot information from it whereas other intensity based segmentation algorithms are unable to generate reliable results due to undistinguishable variations between the background and region of interests.Template matching with cross correlation(CC) [3], Neural Networks(NN) [4,5,6] and Support Vector Machines(SVM) [7] have also been in use for detecting braille dots from the digital image.Though the SVM,CC and NN gives a satisfactory performance in dot detection and character recognition , they still fail to give high accuracy for interpoint braille print or low quality images.The huge processing time for high resolution images especially for CC and training NN is another issue that needs consideration.Figures 7,8 and 10 provides the necessary performance analysis of these methods.This paper reports the use of a composite feature detection algorithm called GFV (Generalized Feature Vector) which successfully detects the dots even for low quality image .

A. Feature Extraction using GFV

The basic idea of the GFV framework is to represent the total feature set $\mathbf{x}$ in terms of a multimodal probability distribution in a multidimensional feature space, mathematically expressed as

$$\mathbf{x} = \{\mathbf{x}_1, \mathbf{x}_2 \cdots \mathbf{x}_n\} \tag{1}$$

where $\mathbf{x}_i \ \ (i = 1...n)$ denote distinguishable features like color,texture,energy,shape,size etc.The concept of GFV and its feature space is elaborated in figure 6 .Some of these features may be orthogonal to the other.In principle GFV can include as many features as desired. Although inclusion of multiple features can improve the detection reliability it however may increase the computation cost. Therefore for optimal performance not more than three features should be used. However the actual number of features will depend on the application requirements and available computational resources. For any object to be represented in a GFV framework it should have more than one feature like color, texture, energy, shape etc.The classification of the object of interest (braille dots for this particular case) is done using the combined distance metrics using all the feature parameters.The combined distance metrics is calculated from the distance metrics of the individual feature parameters obtained using the mahalanobis distance classifier. The parameter estimation $\theta(\theta = \{\mu, \sigma^2\})$ (required for calculating the mahalanobis distance between the distributions) of the sample's distribution is done using the maximum likelihood estimator (MLE) as it is known to give a consistent, convergent unbiased and minimum variance estimate for gaussian distributions, provided such an estimate can be determined for a particular sample.It can be proved that using a multimodal probability distribution ,the distance metrics σ_{comb} obtained by combining all the variables is less than the individual distance metrics $\sigma_1, \sigma_2, \sigma_3$ for each of the variables $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3$ (Lemma 1). It is known that for a multiparametric search space

$$\frac{1}{\sigma_{comb}} = \frac{1}{\sigma_1} + \frac{1}{\sigma_2} + \cdots + \frac{1}{\sigma_n} \tag{2}$$

Lemma 1 can be proved easily from the above identity using proof by contradiction.It is assumed that $\sigma_{comb} > \sigma_1$. Thus

$$\frac{1}{\frac{1}{\sigma_1} + \frac{1}{\sigma_2} + \cdots + \frac{1}{\sigma_n}} > \sigma_1$$
$$\Leftrightarrow \frac{1}{\sigma_1} > \frac{1}{\sigma_1} + \frac{1}{\sigma_2} + \cdots + \frac{1}{\sigma_n}$$

The above identity is wrong for all values of σ_1 as σ_1 is a square root of a squared term and hence is always positive.Thus the above assumption is wrong and it can be stated that $\sigma_{comb} < \sigma_1$. σ_{comb} is similarly less than σ_2, σ_3..... Hence it is seen that σ_{comb} is less than the standard deviations of all the variables and also decreases with the increase in the number of variables.The

(a) Good Quality Scan

(b) Degraded Quality Scan

Fig. 3. Two different quality of Scanned Images of Braille Print is used to verify the effect of image quality on the performance of the detection methods.

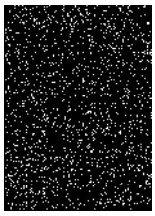


Fig. 4. Application of canny edge detector for detecting dot boundaries from image shown in figure 3b.

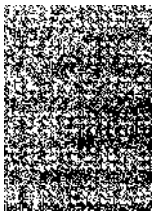


Fig. 5. Application of multilevel thresholding for detecting dots from image shown in figure 3b.

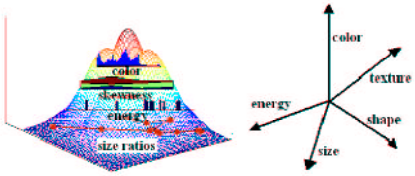


Fig. 6. In statistical sense the GFV can be described as a multimodal probability distribution in a Multidimensional feature space depicted above

efficiency of the multiparametric GFV method over other single parameter based methods is thus established.In the present work the GFV framework includes color, energy-entropy and size ratio features. The feature extraction procedure is further explained in the subsequent subsections.

1) Color: The sample color is represented as the mean and variance of an intensity range $L^c : H^c$ which has probability density function(pdf) value greater than half the maximum pdf value of the sample.

$$I \epsilon L^c : H^c \text{ if } p(I) > \frac{1}{2} \max p(I) \tag{3}$$

$$F_{color} = \{\mu_{L^c:H^c}\ \sigma_{L^c:H^c}\} \tag{4}$$

2) Energy: An weighted average is used to represent the energy of the object $Img_{roi}(m,n,c)$

$$F_{energy} = \sum_{c \epsilon \{red,green,blue\}} \alpha_c \sum_{I=0}^{255} pdf(I,c)^2 \tag{5}$$

where $\alpha_c = \frac{\mu_{Img_{roi}(m,n,c)}}{\sum_{c \epsilon \{red,green,blue\}} \mu_{Img_{roi}(m,n,c)}} . \mu_{Img_{roi}(m,n,c)}$ is the mean of the entire selected region and is given by $\sum_{I=0}^{255} I.pdf(I,c)$.

3) Size ratio: A number of size ratios can be included in the feature vector.Some of them are listed below:

(a) s1:area of the object/area of the minimum convex polygon bounding the object

(b) s2:perimeter of the object/perimeter of the minimum convex polygon bounding the object

(c) s3:area of the object/area of the minimum rectangle bounding the object

(d) s4:perimeter of the object/perimeter of the minimum rectangle bounding the object

$$F_{szratios} = \{s1\ s2\ s3\ s4\} \tag{6}$$

The total feature vector $\overrightarrow{F_{roi}}$ of the model $Img_{roi}(m,n,c)$ can thus be represented by

$$\overrightarrow{F_{roi}} = \{F_{color}, F_{energy}, F_{szratios}\} \tag{7}$$

In the second step, desired object is detected by matching the sample feature vector with the input image.This is accomplished in three steps:(1)Initially the search space is reduced to certain distinct regions $S(reg)$.These regions are obtained by filtering the image such that the output image contains regions which has an intensity that lies in the range of scaled deviation $\delta\sigma_{L^c:H^c}$ of the sample mean $\mu_{L^c:H^c}$ as shown in equation 8.(2)The feature vector for each of the regions is obtained.(3)The best matching is then determined using a distance metrics of the sample vector and region vectors using a selected threshold determined experimentally.

$$I^c \epsilon S(reg, c) \text{ iff } I^c \epsilon \mu_{L^c:H^c} \pm \delta\sigma_{L^c:H^c} \quad (8)$$

GFV reference model is created using a selected dot model.The image is then segmented using the reference color parameter and energy,entropy for each segmented region is calculated.Regions which have a distance metrics less than the threshold value of 0.05 is extracted as dots .The size ratio parameter is required for dot differentiation to identify whether a dot is a merged dot or a single dot(either back side or front side dot).The performance of dot detection using GFV feature extractor is compared with SVM,NN and CC methods and the results are presented in figure 7 to 10.

B. Dot Spacing Measurement

Once the dots are detected,the next task lies in finding its spacing.The dots are represented using their centroid position which is based on the standard center of mass equation in discrete form.In order to find out the inter-dot spacing the dots are first marked in a sequential manner.This is done using the following steps:
(1) The dots are initially clustered using the k-means clustering algorithm where the number of clusters equals the number of lines.Figure 11 shows the clusters.
(2) In each cluster the dots are again grouped into three according to their increasing x-distance.This is further depicted in figure 12
(3) The dots are finally numbered using their y-distance. The final sequence is thus seen in figure 13.Once the numerical arrangement of the dots are complete they can be used to compute the spacings.It is evident from the figure 12 that the x-distance between dots of two groups j and j+1 gives the inter-dot horizontal spacing if j is odd and inter-cell horizontal spacing(i.e. inter-character spacing) if j is even.From figure 13 it is obvious that the y-distance between two dots k and k+1 for each group j gives the inter-dot vertical spacing.

The results of dot spacing measurements computed using the above procedure for two prototypes are shown .It can be clearly visible from the results in table I (showing the horizontal dot spacing HDS,vertical dot spacing VDS and horizontal character spacing HCS) that the earlier model had some ambiguity .This can be further justified from a zoomed area of the print (figure 14) where it can be seen that the center dots are slightly shifted. The prototype was thus modified according to the requirements shown in figure 2. The horizontal and vertical spacings for three samples from the current printing unit are shown in tables II , III and IV.The pixel to metric scale conversion is calculated

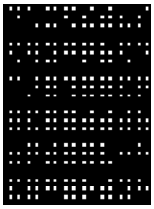
(a) Dot detection using SVM

(b) Dot detection using NN

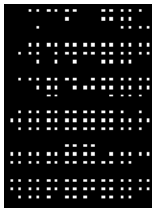
(c) Dot detection using CC matching

(d) Dot detection using GFV feature extractor

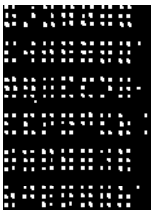
Fig. 7. The dot detection from image in figure 3a using the individual methods are shown.The performance comparison is further verified from the plot in figure 10



(a) Dot detection using SVM



(b) Dot detection using NN



(c) Dot detection using CC matching

(d) Dot detection using GFV feature extractor

Fig. 8. It is evident from the figure that the GFV method is able to give a much more reliable and consistent detection even for a poor quality image.

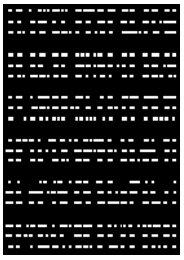
HCS	3.45	3.5	2.6	3.45	3.45	3.5	2.6	3.45
HDS	2.4	2.4	2.4	2.5	2.6	2.4	2.4	2.5
VDS	2.14	2.21	2.14	2.28	2.14	2.14	2.28	2.21

TABLE I
DOT MEASUREMENTS OF IMAGE SHOWN IN FIGURE 14.

using the paper size and image resolution.The tables signify that the measured cell spacings are almost similar to the required spacings shown in figure 2.The minute differences in the second decimal place mainly occurs due to dot area detection and can be ignored.The three samples used for measurement are taken from three different areas of the print.Hence it can be concluded that the spacings are uniform across the entire sheet.

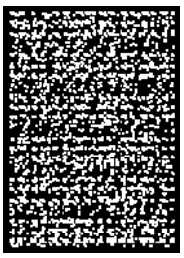
$S1$	2.4464	2.4464	2.4230	2.4230	2.4230
	2.4464	2.4464	2.4464	2.4464	2.4464
$S2$	2.4230	2.4464	2.4464	2.4464	2.4464
	2.4464	2.4464	2.4464	2.4230	2.4464
$S3$	2.4464	2.4464	2.4230	2.4464	2.4464
	2.4464	2.4464	2.4464	2.4464	2.4464

TABLE II
HORIZONTAL SPACING BETWEEN TWO DOTS OF A CHARACTER.



(a) Detection with SVM using possible row positions and all columns of the image.

(b) Detection with NN using possible row positions and all columns of the image

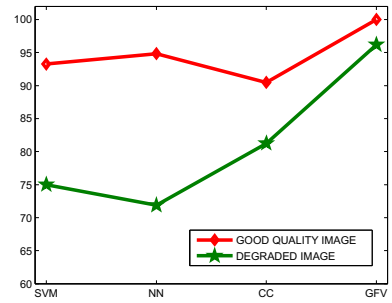


(c) Detection with NN using all rows and columns in the image.

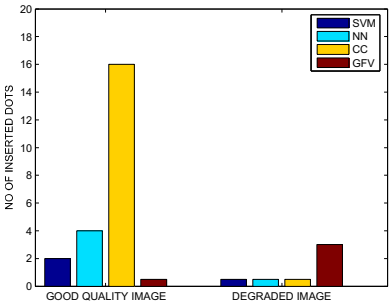
Fig. 9. It can be seen from the figures that both the methods have a lot of inserted dots when used without shifting windows.Thus using the SVM and NN methods for a low quality image without feeding in the possible dot positions becomes absolutely impossible.

$S1$	2.1427	2.1427	2.1341	2.1341	2.1341
	2.1427	2.1427	2.1427	2.1427	2.1427
$S2$	2.1341	2.1427	2.1427	2.1427	2.1427
	2.1427	2.1427	2.1427	2.1427	2.1427
$S3$	2.1427	2.1341	2.1427	2.1427	2.1427
	2.1427	2.1427	2.1427	2.1427	2.1427

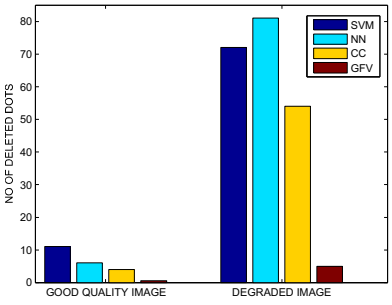
TABLE III
VERTICAL SPACING BETWEEN TWO DOTS OF A CHARACTER.



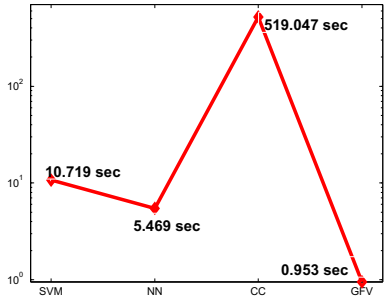
(a) Precision of the different detection techniques.



(b) Total number of inserted dots for the different detection techniques using both the image types



(c) Total number of deleted dots for the different detection techniques using both the image types



(d) Average computation time for detection.Only the testing time for the SVM and NN is considered as training for NN alone takes a few hours depending on the data size.

Fig. 10. Performance Comparison of other popular methods with GFV shows that the later gives a much higher precision and faster processing speed compared to the other methods. The number of deleted dots is much less in case of GFV even for a degraded image.For the degraded image,detection using SVM and NN shows nil inserted dots as only the possible dot positions are checked for the existence of a dot.Results obtained on checking all the image rows and columns generates lot of inserted dots as shown in figure 9 and cannot be used.

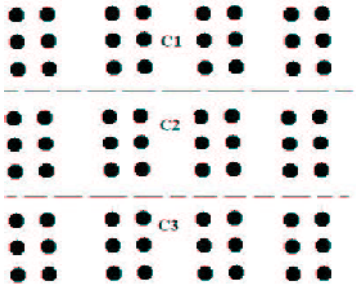


Fig. 11. Figure shows that three clusters C1,C2 and C3 formed from k-means clustering

$S1$	3.5009	3.5009	3.5431	3.5009	3.5009
$S2$	3.5009	3.5009	3.5009	3.5009	3.5009
$S3$	3.5009	3.5009	3.5431	3.5009	3.5009

TABLE IV
HORIZONTAL SPACING BETWEEN TWO CHARACTERS.

III. VISION BASED INTERPOINT BRAILLE CHARACTER RECOGNITION

Vision based character recognition from an interpoint braille print (figure 3a) becomes an intricate task in case of a interpoint print.In general the dot detection algorithm detects both front and back side dots for an interpoint braille print as seen in

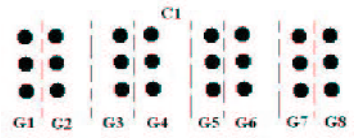


Fig. 12. Figure depicts the grouping of the dots in each cluster according to their x-distance

Fig. 13. Figure depicts the ordering of the dots in each group according to their y-distance

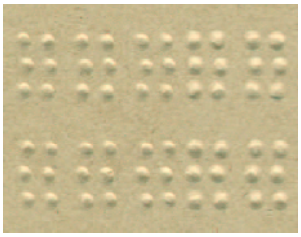


Fig. 14. Figure represents a zoomed portion of print using an earlier prototype

figure 7d.Hence it is essential to distinguish between the front side and back side dots so that the character recognition for each side can be carried out individually.Section III-A presents a simple gridding method to perform the dot differentiation.Code generation for each side dots are then carried out as described in section III-B.

A. Front and Back Side Dot Differentiation

The general structure of the interpoint dots can be seen from the figure 15. The upper and lower set of dots denote the back side and front side braille dots respectively.These dots are differentiated as front or back sided by a griding method as illustrated using figure 15.The detection window is chosen accordingly for the front or back side as shown in the figure.For both the cases,dots whose top and bottom extremes lie above and below the lines R_1,R_2 and R_3 are accepted and their centroids (d_{row}, d_{col}) are noted.If any of the accepted dot is a mixed dot(i.e the combination of a back and front dot) then its centroid is updated using the algorithm given below.

1.for each detected dot

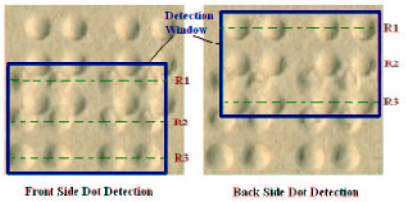


Fig. 15. The figure shows the detection window and intersecting lines R_n to be considered for front and back side dot distinction.

2.if R_n intersects the dot
3.{
4.if difference of d_{row} and row coordinate of R_n is greater than a threshold
5.{
6.set d_{row} to row coordinate of R_n
7.set d_{col} to average of d_{col} and left extreme of dot
8.}
9.}

The above algorithm detects the front side dots.For detecting back side dots line 7 of the algorithm needs to be changed as set d_{col} to average of d_{col} and right extreme of dot
The threshold to be selected depends on the image resolution.For the present work it is set to 3.

B. Code Generation

A sliding window with a fixed interval is used to slide over the entire braille image to detect each braille cell.This operation does not give any ambiguity as it is already verified that the inter-dot and inter-cell spacings are fixed. Thus after detecting the front side or back side dots,a starting window and an interval width is selected.The window then slides over the entire image detecting each cell.Each cell is divided into grids as shown in figure 16 and the corresponding code for the cell is generated according to the presence or absence of a dot centroid in each grid following the configuration.

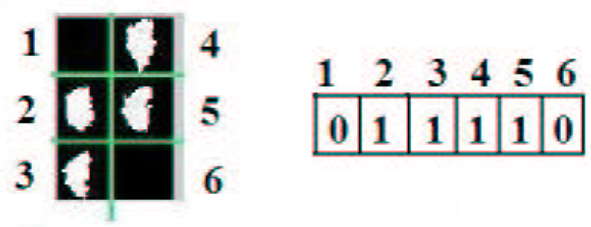


Fig. 16. Figure shows griding of each cell and the cell numbering order used for code generation

English word reconstruction is then carried out from the generated binary code using a lookup table.Table V and VI

shows an example of the generated code and corresponding English conversion of the front and back side dots of figure 3a.

110000	111010	100000	010100	111000	111000
B	*R*	*A*	*I*	*L*	*L*
100010	000000	001000	100000	000000	110000
E		'	*A*		*B*
111000	010100	101110	100110	000000	110100
L	*I*	*N*	*D*		*F*
111010	100010	101110	100100	110101	101100
R	*E*	*N*	*C*	*H*	*M*
100000	101110	000101	000001	100010	100000
A	*N*	.	*CAPS*	*E*	*A*
100100	110010				
C	*H*				

TABLE V
GENERATED CODE AND CORRESPONDING ENGLISH CONVERSION FOR FRONT SIDE

101110	101001	101100	110000	100010	111010
N	*U*	*M*	*B*	*E*	*R*
100010	100110		010000		011110
E	*D*		,		*T*
101010		010010	000001	110100	111010
O		:	*CAPS*	*F*	*R*
101010	101100		011110	101010	111100
O	*M*		*T*	*O*	*P*
011110	101010		110000	101010	011110
T	*O*		*B*	*O*	*T*
011110	101010	101100		100010	101110
T	*O*	*M*		*O*	*N*

TABLE VI
GENERATED CODE AND CORRESPONDING ENGLISH CONVERSION FOR BACK SIDE

IV. DISCUSSION AND CONCLUSION

The experimental results verify that the performance of the GFV approach in detecting the braille dots is greater than other conventional approaches.The potential of the feature extractor can be judged from the fact that it gives a high accuracy even for low quality images where other methods perform poorly.This enables the quality assessment procedure to be achieved with greater ease and accuracy.A simple but elegant approach to convert the braille code to running text is also proposed.The binary code generator for each Braille cell used for word reconstruction provides satisfactory result with a high accuracy and very low processing time. The character recognition from interpoint braille print works with a 100 percent accuracy, successfully distinguishing between front side and back side dots and generating the English text for both sides of the page simultaneously .The English text provided for printing and the reconverted English output can be compared to verify the accuracy of the braille coding system used during printing.Besides it can also be utilized in regenerating braille print from existing ones.

REFERENCES

[1] Saad D Al-Shamma and Sami Fathi,"Arabic Braille Recognition and Transcription into Text and Voice", 5th Cairo International Biomedical Engineering Conference Cairo, Egypt,2010

[2] AbdulMalik S Al-Salman, Ali El-Zaart, Yousef Al-Suhaibani, Khaled Al-Hokail, AbdulAziz O Al-Qabbany,"An Efficient Braille Cells Recognition", IEEE 2010

[3] Jan Mennens,Luc Van Tichenens,Guido Francois, Jan J Engelen,"Optical Recognition of Braille writing using standard Equipment",IEEE transactions on Rehabilitation Engineering,1994.

[4] A. Antonacopoulos and D. Bridson,A Robust Braille Recognition System,Document Analysis Systems VI, Springer Lecture Notes in Computer Science, LNCS 3163, 2004, pp. 533-545.

[5] Giovanna Morgavi, Mauro Morando,A neural network hybrid model for an optical braille recognitor.

[6] Lisa Wong Waleed Abdulla Stephan Hussmann,A Software Algorithm Prototype for Optical Recognition of Embossed Braille

[7]Jie Li, Xiaoguang Yan,Optical Braille Character Recognition with Support-Vector Machine Classifier,2010 International Conference on Computer Application and System Modeling (ICCASM 2010)