2015 International Conference on Industrial Instrumentation and Control (ICIC)
College of Engineering Pune, India. May 28-30, 2015

Dot Matrix Text Recognition for Industrial Carton Classification

Siddharth Nitin Patki [1]
Dr. Prof. Madhuri Joshi [2]
Dept. of Electronics and Telecommunication Engineering
College of Engineering Pune, Pune, India
[1]sinipat@gmail.com, [2]maj.extc@coep.ac.in

Abhishek Ninad Kulkarni [3]
Dept. of Electronics and Telecommunication Engineering
Vishwakarma Institute of Technology, Pune, India
[3]abhi.bp1993@gmail.com

***Abstract*— Automatic classification of packaging cartons according to their contents is an industrial need. In this paper we present an Optical Character Recognition (OCR) system to segment and recognize the sparse dot matrix text printed on the cartons in order to classify them based on the contents. Proposed solution is robust to non-uniformities in background illumination, shadow artifacts, inclined text, degraded text due to missing dots etc. We propose efficient segmentation technique using simple morphological operations which makes use of the discrete nature of the dot matrix text in distinguishing it from other information. The dot matrix characters can be uniquely characterized by analyzing the pattern of dots. We retrieve this pattern, and feed it as feature vector to the trained Support Vector Machine (SVM) classifier. The combination of the unique patterns and SVM classifier results into high character recognition accuracy, in turn leading to efficient carton classification. Finally, we discuss the result statistics of character recognition and carton classification.**

Keywords*— *Dot Matrix Text Recognition; OCR; Dot Matrix Text Segmentation; Industrial Carton Classification.

I. INTRODUCTION

A dot matrix printed character is composed of discrete dots printed in specific order. Dot matrix printing is an extremely cheap alternative to the high quality inkjet printing, and is generally used when the printed content is more important than the print quality. Due to their cost effectiveness, they are extensively used in packaging industries in India to print the package contents on the cartons. Thousands of cartons are processed every day in an industry, and the information printed on them varies a lot according to the contents. Manual classification of such high number of cartons, and keeping trac of them is a tedious job. We propose a robust OCR system to segment and recognize the sparse dot matrix text printed on cartons in order to classify them automatically. Other information printed on the cartons can also be read, and stored in the central database for record.

The discrete nature of the dot matrix text renders the standard character recognition techniques ineffective. Reconstructing the characters by connecting the dots does not prove to be very effective as well. In addition, the text could be degraded in some cases, making the task more challenging.

For each dot matrix character, the dots are printed in specific pattern. We seek to retrieve this pattern and use it as a feature for classification. The algorithm is tested in cases having non-uniform background illumination, shadow artifacts, inclined text, damaged cartons, blur images etc. proving its robustness.

For the effectuation of the proposed approach, rest of the paper is organized as follows: Prior work is discussed in section II. System flow and detailed working of each block is explained in III. The experimentation and results are presented in section IV. We then conclude our paper in section V.

II. PRIOR WORK

A pitch based approach to segment dot matrix text has been suggested by Berrin in 2000 [1]. A major drawback of the approach is that it assumes the dot matrix text has fix pitch (character width). It cannot be used in the case where the dot matrix characters have varying widths. In addition, their approach is computationally expensive. A principle component analysis based approach to differentiate dot matrix printed documents from high quality printed documents is proposed by Sachdeva and Gard in 2010 [2]. It does not offer solution for the dot matrix text segmentation

III. SYSTEM FLOW

Fig. 1 shows the flow of operations to be performed in our framework.

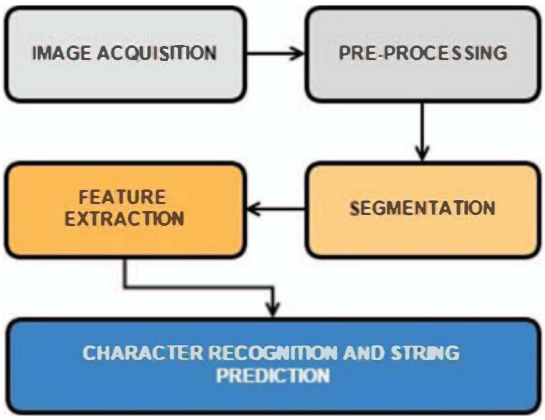

Fig.1 System Flow Diagram

A. Image Acquisition

Fig. 2 demonstrates the proposed industrial setup, and the approximate location and type of printed information.

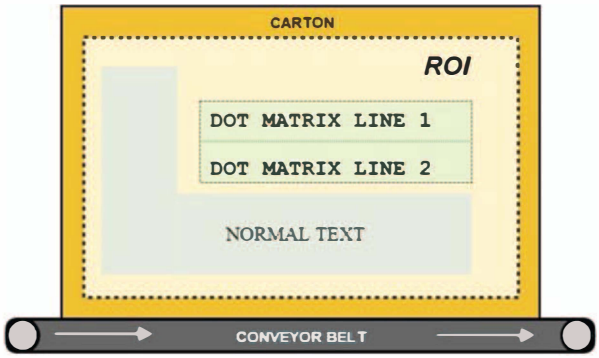

Fig.2 Proposed Industrial Setup and details of printed information.

The camera mounted in front of the conveyer belt detects the carton, captures the image and automatically crops the region of interest (ROI). For experimentation, we captured the still images using a camera having 12 MP resolution, and manually cropped the images to obtain ROI. The images were captured in varying lightning conditions, and from varying distances and angles for experimentation. Few images of abnormal cases e.g. inverted printing, high blur or physically damaged cartons were also included in the database to test the system adaptability. A sample image is shown in Fig. 3

Fig.3 An actual carton image showing dot matrix text.

Following problems were observed in the images:

- Inclined text
- Non uniform lighting or shadow artifacts
- Motion Blur

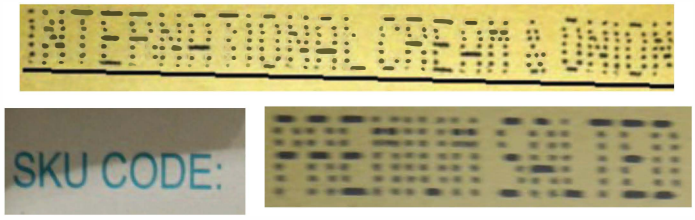

Fig.4 Images showing inclined text, shadow artifact and motion blur.

B. Pre-Processing

Preprocessing is required to convert the original RGB image into its binary equivalent, while preserving the desired information. We resize the RGB image to 15 % of its original size to reduce the computations required. The averaging operation performed during resizing also helps to remove the salt and pepper or Gaussian noise. To deal with non- uniform background illumination and shadow artifacts, we use bottom-hat filter [8] on the grayscale image.

1) Bottom-Hat filter

In case of grayscale images, the closing [8] operation, enhances the local minima (small dark regions) while not affecting the global minima and maxima. Thus closing can be used to compensate for non-uniform background illumination when the image has lighter background and darker foreground (characters). It can be seen in Fig. 3, that the background is darker towards left and upper regions of the image. The uneven illumination makes image thresholding difficult. Fig. 5 (b), for example, is a thresholded version in which the characters in the brighter region are well separated from the background while the characters in the dark region are lost. Closing of the image can produce reasonable estimate of background across the image, as long as the structuring element is large enough not to fit entirely in the local minima (characters), Fig. 5 (a).

The bottom-hat filter is defined as the closing of the image minus the image itself. $(f \bullet k) - f$. Thus in our case the carton image is subtracted from its approximated background leaving out the foreground characters. It is then converted to binary form using Otsu's automatic threshold selection algorithm [3], Fig. 5 (c). This process makes images consistent, and ready for text segmentation.

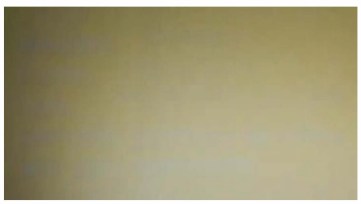

Fig.5 (a) Background approximation obtained after closing the carton image using disk shaped structuring element.

(b) Binary image obtained before using Bottom-Hat filter.

(c) Binary image obtained after using Bottom-Hat filter.

C. Segmentation

Identifying the region of image containing the textual information is needed in order to extract the individual characters. Higher the accuracy of segmentation better will be the character recognition. The task is divided into two parts.

- Line Segmentation:

The discrete nature of the text demands for a non-conventional text segmentation technique. The usual methods such as connected component analysis, sliding window, or horizontal and vertical projections based techniques which assume the character wholeness, cannot be directly used for segmenting the dot matrix characters.

We use the discrete nature of the dot matrix text to our advantage in devising solution to the problem of segmentation. A group of connected pixels in an image is called as an object. Dot matrix text is an ordered arrangement of such small objects. The number of pixels associated with an object is called as its area. A flood fill algorithm [4] is used to detect objects in the image. Then the objects having area less than a particular threshold are filtered. The resulted image is then subtracted from the original image to get the cluster of dot matrix text as shown in Fig 6.

Resulted image has the clusters of dot matrix text. However, undesired objects having area similar to the dots are also filtered. Thus further processing is required to segment out final text. As any text is generally spread in the horizontal direction, we perform dilation [8] on the image using a rectangular element having width 8 times its height. This way the clusters get connected, and form 2 big objects as shown in Fig. 6. Smaller rectangular objects belong to the non-textual information. We again use the flood fill algorithm to calculate the area of objects and select the top two highest area objects and find their bounding boxes. The original binary image containing full dot matrix text is then cropped using calculated bounding boxes to extract the text.

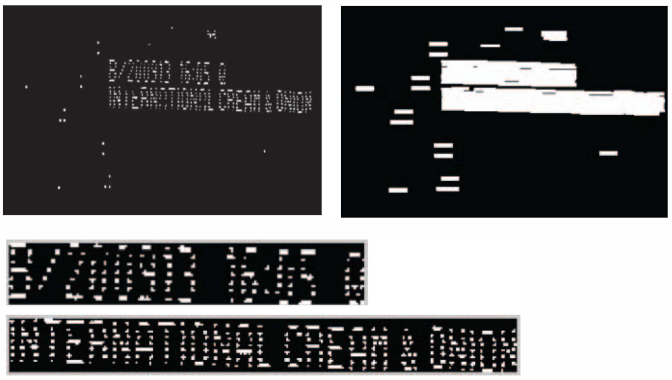

Fig.6 Filtered dot matrix cluster, Result after using dilation, and segmented dot matrix lines respectively.

- Character Segmentation:

In order to recognize the characters, they must be extracted from the segmented dot matrix lines. We use vertical projections for character segmentation.

Vertical projections can be simply defined as a row vector, containing the number of white pixels present in each column.

$$\boldsymbol{VP(c)} = \sum_{\boldsymbol{r}} \boldsymbol{i(r,c)}$$

As the characters are separated with a blank space in-between, the vertical projections in that region are zero. We use this idea to segment each character. Before calculating the projections, we first dilate the image using vertical element to increase the character mass, so that a clear distinction can be made between high and low projection values. Fig.7 demonstrates the thresholded projections of dilated image.

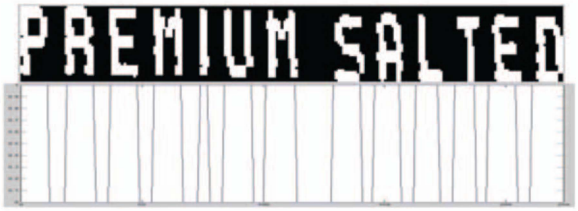

Fig.7 Thresholded vertical projections of joined dot matrix text representing characters and gaps in-between.

Projections also give us the pitch of the characters. It can be effectively used to easily distinguish characters such as 'I' which has small pitch (width).

Each dot matrix character is cropped and finally the character image is resized to the size of 50 x 70 pixels.

D. Feature Extraction

As the name suggests, a dot matrix character is a specific arrangement of dots printed in a matrix. Thus the dot matrix printed characters can be uniquely characterized by analyzing the arrangements of dots. On observation we found that, the characters in our case have following properties:

- There are maximum 7 discrete dots in Vertical direction and 3 in horizontal direction.
- In the characters having horizontal lines, as shown in Fig.8 below, the dots overlap on one another forming a bigger object.
- We quantify the spread of these objects as approximately 3-dots wide or 6-dot wide. As shown in Fig.8, the top or bottom elements of 'D', 'T' have wider span than top or bottom elements of 'O'. This information is useful in distinguishing between characters such as 'O' and 'D' which are morphologically similar.

Fig.8 Segmented dot matrix characters 'D', 'O', and 'T'

1) Centroid Calculation:

To extract the unique arrangement of dots in a character, we first calculate the centroids of each connected components (dots) in the character image. The dots in the character image are identified using flood fill algorithm, and then their centroids are calculated.

2) Feature Vector generation:

As shown in Fig. 9 the character image is equally divided into 7 vertical and 3 horizontal (a 7x3 matrix) parts. Then the cells of matrix containing dots are identified by checking the presence of centroid within their bounding coordinates. Assigning '1' to occupied cell and '0' to the unoccupied, we get a feature vector of 21 instances.

When the centroid is detected in the middle column its spread is determined and only if it is found to be greater than 3-dot (35 pix) spread, the complete row is marked '1' (1-1-1). This provides an elbow-room for distinguishing between O and D as shown in Fig. 10.

Fig.9 Segmented character 'P' showing centroids locations of each dot; Imaginary grid dividing the image into 21 cells; Feature image obtained after sensing the arrangement of dots respectively.

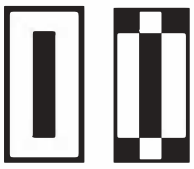

Fig.10 Feature image of characters 'D' and 'O' showing clear distinction.

3) Correction to Irregularities:

It may be possible that there are three different but displaced dots in a horizontal line resulting into (1-1-1) row values. In such case, the instance sequence is replaced with (0-1-0) to differentiate it from a horizontally spread single object producing (1-1-1) sequence occurring in characters such as 'D' or 'P' etc. Fig. 11 shows the character 'N', and its feature vector before and after connection. This correction enables us to distinguish between characters such as 'N' and 'H'.

Fig.11 Segmented character 'N' alongside its feature image before and after correction.

E. Classification

We use Support Vector Machine (SVM) as a classifier.

1) Introduction to SVM:

SVM finds the maximum margin hyperplane for classifying the labelled training data points by visiting a higher dimensional space. The margin can be defined as the smallest of all the Euclidian distances of data points from the decision boundary. Maximizing the margin over the training data generally results into better generalization. Given training vectors $x_i \in R^n$, $i = 1, \ldots, l$, in two classes, and an indicator vector $y \in R^l$ such that $y_i \in \{1, -1\}$, C-SVC (Support Vector Classifier) (Boser et al., 1992; Cortes and Vapnik, 1995) solves the following primal optimization problem to find an optimal hyper-plane for non-linear patterns.

$$\begin{aligned} \min_{\boldsymbol{w},b,\boldsymbol{\xi}} \quad & \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + C\sum_{i=1}^{l}\xi_i \\ \text{subject to} \quad & y_i(\boldsymbol{w}^T\phi(\boldsymbol{x}_i) + b) \geq 1 - \xi_i, \\ & \xi_i \geq 0, i = 1, \ldots, l, \end{aligned}$$

Where, $\varphi(x_i)$ maps x_i into a higher-dimensional space. Furthermore, $K(x_i, x_j) = \varphi(x_i)^T \cdot \varphi(x_j)$ is called as the Kernel function. Since in our case the data can be separated linearly, we have used the linear kernel, and it is mathematically given by,

$$K(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^T\mathbf{x}_j$$

Sometimes in practice, greater margins can be achieved by allowing the classifier to misclassify some anomalous data points. The parameter ξ in the inequality allows for such errors, and is called as slack variable. $1 \geq \xi \geq 0$ corresponds to an example falling inside the margin, whereas $\xi > 1$ corresponds to a misclassification. The sum of slack variables is a bound on the number of misclassified examples, and thus a regularization parameter C (> 0) is used to control the tolerance. This formulation is called as the soft-margin SVM. Large values of C may lead to over fitting, while small value may lead to under fitting of the data.

We have used an SVM library called as LIBSVM [6] for the implementation of C-SVC. For multiclass classification, C-SVC uses one vs. all strategy.

SVM is trained with linear kernel using a labeled feature set having one or more feature vectors per character. Ideal feature instance sequence can be used while training, but adding more examples per class, especially of the broken characters strengthens the classifier. SVM classifies the unknown character into the closest matching class, and returns the class label. The character represented by that label is then displayed.

2) Decision Phase:

After recognizing the first 6 characters of the dot matrix text describing the package contents, they are compared with the pre-stored labels on the character by character basis, as shown in the table 1. When the guessed character matches the pre-stored character, the score associated with that label is increased by a point. Finally the label with highest score is returned, and carton is classified. If the score repeats itself for two different labels, then further character recognition and comparison is in order. (G.C.: Guessed Characters).

Table 1. Comparison between guessed characters and stored labels

G.C.	PRESTORED LABELS	SCORE
TANGLE	PREMIUM SATLED	0
TANGLE	INTERNATIONAL CREAM & ONION	0
TANGLE	MAD ANGLES MASALA MADNESS	1
TANGLE	**TANGLES** MASALA	**6**
TANGLE	TEDHE MEDHE MASALA TADKA	1
TANGLE	GALATA MASTI SOUTHERN TWIST	1

Text printed in high quality (non- dot matrix) can also be easily recognized by first segmenting the characters using component analysis [4], and then comparing their correlation matrices with the stored database. This information can be appended to database for further use.

IV. Results and Experimentation

System performance was tested on a database of 60 images which included 6 different variants. Images having high blur, inverted printing, damaged carton surface, shadow artifacts were also included in the testing set.

- Text segmentation accuracy:

The dot matrix text was fully segmented in 90 % images, and it was partially segmented in the remaining 10 % images. Partially segmented text had enough characters present to aptly classify the cartons. Inclined text was the prime reason affecting segmentation.

- Character recognition accuracy:

The character recognition accuracy is compared for three classification techniques.

- Euclidian Distance Classification
- Correlation Coefficient Comparison
- SVM

Fig. 12 shows the results obtained for the three techniques on a set of 710 characters having 18 variants. Highest accuracy was obtained for SVM classifier.

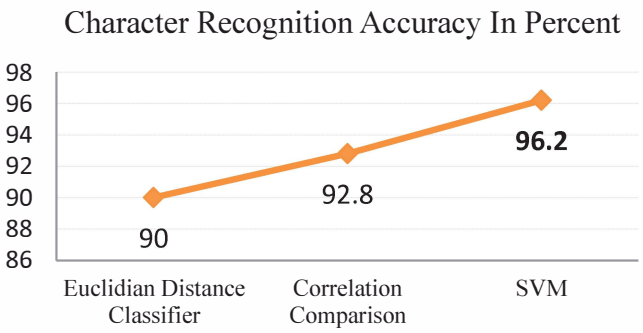

Fig. 12 Character recognition accuracy obtained for three techniques.

- Carton classification efficiency:

59/60 cartons were correctly classified. The misclassified image had highly damaged carton surface obscuring the text.

Fig. 13 shows the abnormal cases which were correctly classified by the algorithm.

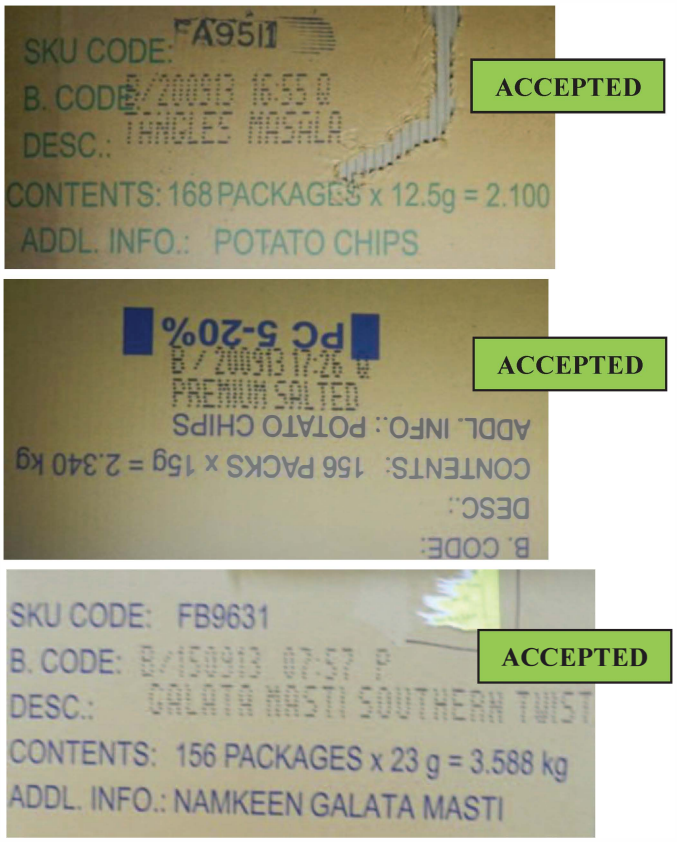

Fig.13 Anomalous cases which were successfully classified by the algorithm proving its adaptability.

All the algorithms were implemented using MATLAB 13 on a machine using Intel Core i5 processor running at 2.3 GHz clock speed.

V. CONCLUSION

A novel OCR system is proposed to segment and recognize the sparse dot matrix text printed on industrial carton. Classification was tested for 6 variants, and the cartons were classified with high consistency and efficiency. The system performance was found to be promising for the anomalous cases, proving the adaptability of the system. A novel approach is presented for accurately and speedily segmenting the sparse dot matrix text. The segmentation was fairly efficient in spite of non-uniform background illumination, shadow artifacts, inclination, blur etc. An approach to recognize the dot matrix characters by analyzing the arrangement of dots is proposed. A method identifying the convention used by the printer itself while printing different characters leads to a highly efficient OCR engine. The simple and effective techniques were the result of close observation and understanding of the nature of the information we dealt with.

Acknowledgment to BARC grant on "Palm print and Finger print recognition".

VI. REFERENCES

[1] Berrin A. Yanikoglu, "Pitch based segmentation and recognition of dot matrix text", *Springer International Journal on Document analysis and recgnition*, pp. 34–39, 2000.

[2] Sachdeva, M.; Garg, U., "Detection of Dot Matrix Printed Documents Using Component Analysis," *Advances in Computer Engineering (ACE), 2010 International Conference on* , vol., no., pp.125,129, 20-21 June 2010.

[3] Otsu, N., "A Threshold Selection Method from Gray-Level Histograms," *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 9, No. 1, 1979, pp. 62-66.

[4] Haralick, Robert M., and Linda G. Shapiro, "Computer and Robot Vision", Addison Wesley, vol. 1, pp. 28-48, 1992.

[5] Lam, L., Seong-Whan Lee, and Ching Y. Suen, "Thinning Methodologies-A Comprehensive Survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol 14, No. 9, September 1992, page 879, bottom of first column through top of second column.

[6] Chih-Chung Chang and Chih-Jen Lin, LIBSVM: a Library for support vector machines. ACM Transactions on Intelligent Systems and Technology, 2:27:1--27:27, 2011. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm

[7] Babu, D.R.R.; Ravishankar, M.; Kumar, M.; Raj, A.; Wadera, K., "Recognition of machine printed broken characters based on gradient patterns and its spatial relationship," Computer Science and Information Technology (ICCSIT), 2010 3rd IEEE International Conference on, vol.1, no., pp.673,676, 9-11 July 2010

[8] Gonzalez, R. C., R. E. Woods, and S. L. Eddins, Digital Image Processing Using MATLAB, Gatesmark Publishing, 2009