OCR for Unreadable Damaged Characters on PCBs Using GSC Algorithm and kNN Classifier

Carlos F. Nava-Dueñas, *member IEEE*
Skyworks Solutions, Inc.; Engineering Institute
UABC, Mexicali, Mexico
nava.carlos@uabc.edu.mx

Felix F. Gonzalez-Navarro, *member SMIA*
Engineering Institute
UABC, Mexicali, Mexico
fernando.gonzalez@uabc.edu.mx

Abstract— **In this paper, we propose to change the actual implemented pattern matching method to have optical character recognition by implementing the Gradient, Structural, Concavity (GSC) algorithm to extract the features of damaged, unreadable or incomplete numerical digit characters from images on printed board circuits (PCBs). Grayscale color images are acquired from a charge-coupled device (CCD) camera, assembling a dataset of 500 matrix images samples for the character digits from 0 to 9. The GSC feature extraction method is applied to get the characteristics that will be used in the character recognition step. Experimental results show that applying GSC algorithm to extract the features and using k-Nearest Neighbor (kNN) Classifier with the Euclidian Distance can improve optical character recognition (OCR) detectability of damaged characters from actual 95% to more than 97% in early tests.**

***Keywords*— machine vision, optical character recognition (OCR), kNN Classifier, GSC, pattern recognition, printed circuit board (PCB).**

I. Introduction

Optical character recognition (OCR) has been an important technology used to convert characters from a digital image to a digital text. There are basically two types of OCR algorithms: the first technique is related with the matching of matrix images, where an alphabet of stored character images is used to compare with an input image [1], [2]. This pattern matching does not work well when new fonts are encountered or input character images are unreadable. The second technique decomposes an input image to extract the principal features [3], [4], [5]. Then, classifiers are used to compare the input image features with some stored image features and choose the best match.

Our actual system implemented at the Skyworks factory uses the traditional OCR technique i.e. pattern matching. Our implemented vision system reads identification characters on printed circuits boards (PCBs) for lot integrity and machine control. This commonly-used technique is not robust enough because many of the images on PCBs present some damaged characters due to dirt or as a result of bad previous processes [1]. Actual OCR detectability is around 95% at best. It starts with a monochrome VGA image acquisition of the upper left section of a PCB, using a NI-1752 smart camera, with full resolution, 640x480 pixels and maximum data transfer @60 fps using a GigE port. The selected resolution and data transfer speed parameters meet the factory production schedule of inspected PCBs. The camera has a grayscale output image type with a maximum character resolution to cover the entire PCB characters positions, as shown in Fig. 1.

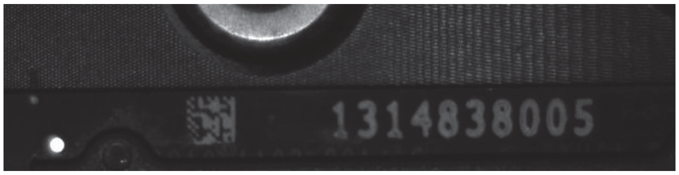

Fig. 1. PCB with no damaged characters.

Due to some problems with previous processes in the production line, some PCBs will have some residual dirt over the characters, making some characters unreadable for the pattern matching technique, as shown in the following Fig. 2.

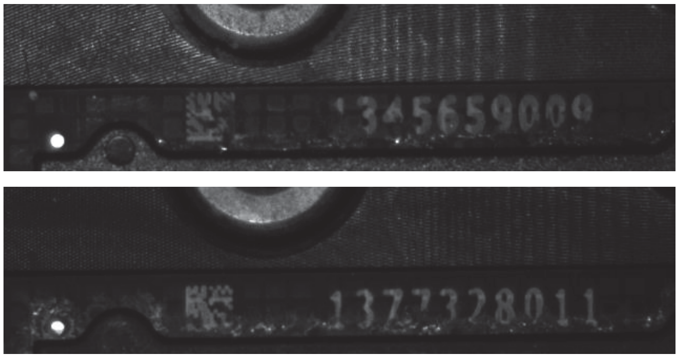

Fig. 2. PCBs with evident residual dirt over characters.

The principal problem is that operators have lower throughput than automatic OCR software, and this leads to manually writing down the information from the screen when the actual recognition software fails, increasing the process time, making possible errors from wrong readings, resulting in higher production costs. Taking into consideration these facts, a better approach has to be considered [3].

This paper presents a proposal for implementing a character recognition technique for unreadable characters using Gradient, Structural and Concavity (GSC) extraction features and K-Nearest Neighbor Classifier using Euclidian Distance [6], [7], [8].

II. Data Set

The experimental data set consists of 500 character images, 50 images correspond to each numerical digit from 0 to 9. Fig. 3 shows some examples of damaged digit image samples.

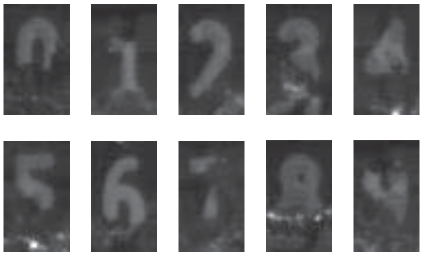

Fig. 3. Some damaged digit images from dataset

For our previous dataset, a pre-processing step is applied as follows:

Let I_i be any digit image of size (k,l) -see Fig.4- from the original dataset, $\forall I_i$:

.

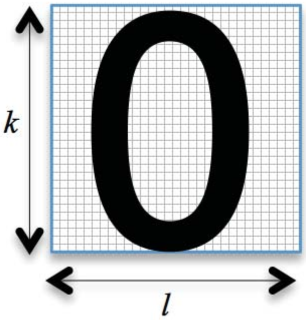

Fig. 4. I_i digit image matrix with size (k,l). k=50 and l=30 in our experiments

1. *Convert to gray-scale (if previous images are color RGB type).*
2. *A threshold is applied to binarize.*
3. *I_i is split in 60 non-overlapping regions (10 x 6 grid), as shown is the Fig. 5.*

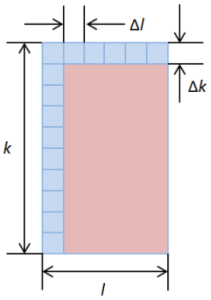

Fig. 5. I_i. is split in a 10 x 6 grid of $\Delta k = \Delta l = 5$ pixels

III. Gradient, Structural and Concavity (GSC) Recognition Algorithm

The GSC algorithm to extract information from the image was implemented [3]. It *constructs* features of an image by applying a three-step feature extraction process: 1-*Gradient step* detect local features by analyzing the stroke shape on small distance; 2-*Structural* step, extract features from stroke trajectories by extending distances of gradient; 3-*Concavity* analysis detects stroke relationships across the image. Fig. 6 shows the final Total Feature Vector (TFV) constructed for each image.

10 x 6 x 12 = 720 Bits	10 x 6 x 12 = 720 Bits	10 x 6 x 8 = 480 Bits
Section I Gradient Features	Section II Structural Features	Section II Concavity Features

Fig. 6. Total Feature Vector of each I_i digit image with size of 1920 pixels

Section I - Gradient Features. Two dimensional convolutions in the X and Y direction is applied to get the gradient features using an 3 x 3 Sobel operators on the original I_i binary image, -see Fig. 7. Gradients from an image representation of the Digit Character 9 are shown in Fig. 8.

$$K_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \qquad K_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

$$G_x = K_x * M \qquad G_y = K_y * M$$

Fig. 7. G_x and G_y are the 2D convolution of a 3 x 3 matrix for every pixel on original I_i digit image matrix. An extra zero padded border is added to I_i.

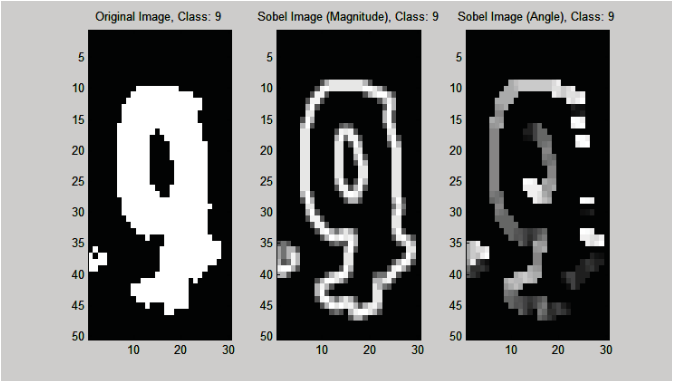

Fig. 8. Gradient magnitude and direction are shown for Digit Image 9

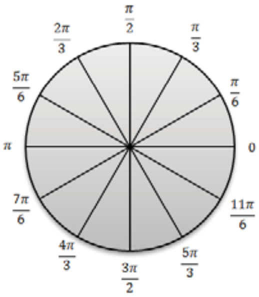

Fig. 9. Gradient range from 0 to 2π in 12 equal space regions

A histogram is applied for each of the 60 non-overlapping regions of the complete image grid, incrementing the counter for every Gradient Angle that falls in each region, as indicated in Fig. 9. A threshold is applied and a final 720 Bits is created for the first part of the TFV, as previously shown in Fig. 6.

Section II - Structural Features. For each pixel of the expanded I_i digit image with zero padded border, a set of 12 rules is applied using 8 pixels around the main pixel. These rules look for specific gradient patterns form with the nearest pixels, like horizontal lines (0, 4), vertical lines (2, 6), diagonals [(5, 1), (3, 7)] and corners [(0, 2), (2, 4), (4, 6), (6, 0)]. Fig. 10 shows these rules in a graphical positioning:

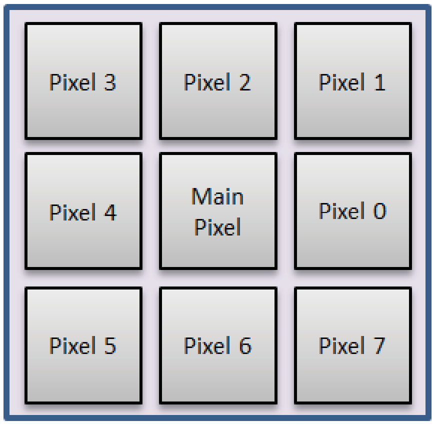

Fig. 10. Eight nearest pixels around Main Pixel

A threshold is applied for each of the 12 rules result, for each of the 60 non-overlapping regions to binarize the complete set. A final 720 Bits set is created for the second part of the TFV, as previously shown in Fig. 6.

Section III - Concavity Features. Three feature sections form the last part of the GSC algorithm:

1) **Coarse Pixel Density:** A histogram is applied to count all the character pixels at each of the 60 non-overlapping regions. Then, a threshold is applied to binarize the result. For this, a 60 Bits new set is included as the first part of the Concavity Features.

2) **Large-Stroke:** Like the previous section, two histograms are applied, one for the horizontal and one for the vertical pixels strokes in each direction. A threshold is applied to binarize the result. For this section, a 2 x 60 = 120 Bits new set is included as the second part.

3) **Upward, Downward, Left, Right and Holes:** In this last section of the concavity features, for every pixel in each of the 60 non-overlapping regions, rays are fired to hit character pixels, borders and check if we have holes or character pixels in all directions. In this last section, a 5 x 60 = 300 Bits new set is included.

As shown in Fig. 6, the TFV for the Section III has the following Bits set:

$$1\,x\ \ (10\,x\,6) + 2\,x\,(10\,x\,6) + 5\,x\,(10\,x\,6) = 480\ \text{Bits}$$

IV. K-NEAREST NEIGHBOR CLASSIFIER

The k-Nearest Neighbor (kNN) Classifier Algorithm was chosen for the damaged character recognition step. It was selected because of its simplicity and fast performance, and the absence of *prior* assumptions about data set probability distributions. The classification occurs when a majority vote among kNNs with respect to any particular test set is given. In the complete experiments k parameter was fixed to 3.

The resulting *Total Feature Matrix* has the form as shown in Fig. 11 –see *Class* is added at the end column:

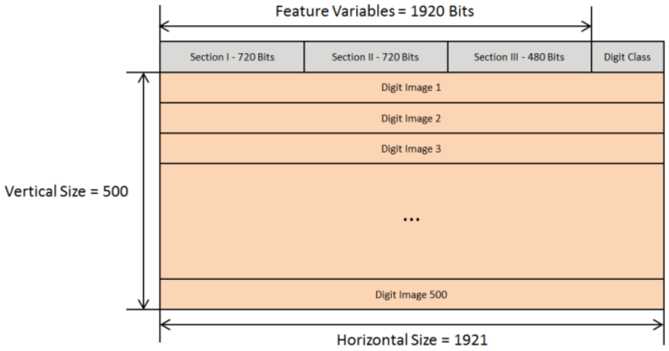

Fig. 11. Final Total Feature Matrix for all Image Dataset with Digit Class included (0 – 9)

The Euclidian Distance was chosen as the first approach to compute the distance metric as follows:

$$D_{Euclidian} = \sqrt{\textstyle\sum_{i=1}^{l}(Test_i - Training_i)^2} \qquad (1)$$

V. EXPERIMENTAL RESULTS

A first-round experiments were conducted as follows: a kNN Euclidian Distance and a K = 3 was trained and tested varying the proportion of training-test samples. A range of 10% to 90% training and 90% to 10% test sets were analyzed. Fig. 12 shows this experimental setting result. Fig. 13 shows a zoom of the 60% to 90% section.

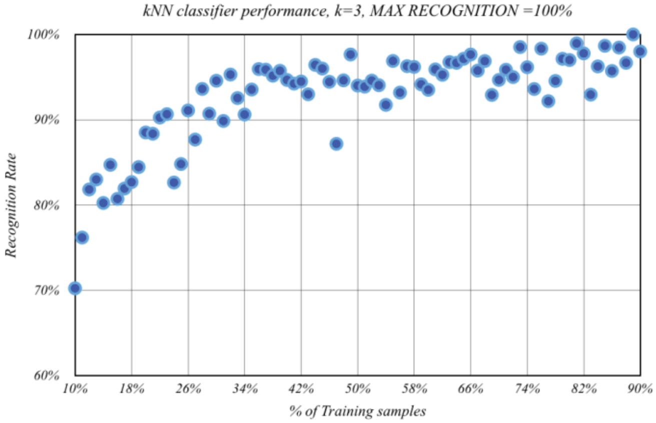

Fig. 12. kNN Classifier performance for all image dataset

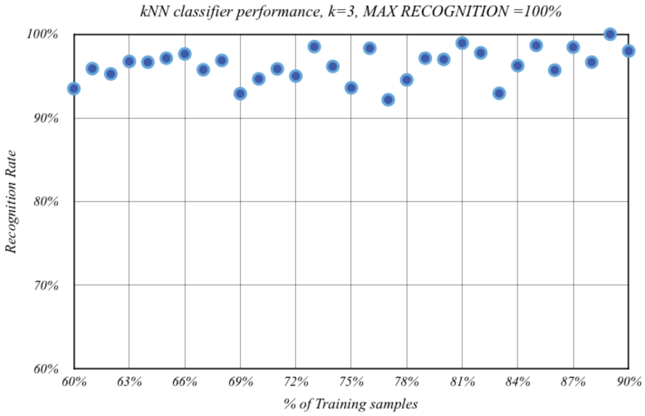

Fig. 13. Zoom in the kNN Classifier performance.

It’s seen that classification stage yields promising results. The kNN Classifier shows interesting peaks of 100% recognition rate at high training percentage by using the GSC algorithm.

In order to assess more precisely the classification performance of the GSC+kNN proposal, Monte Carlo cross-validation strategy was selected [9]. A total of 100 random data splits of 90% training and 10% test samples were analyzed. Fig. 14 shows results and main statistics.

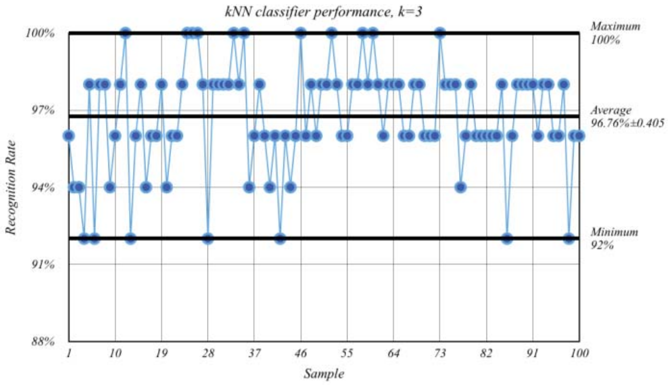

Fig. 14. Monte Carlo 100 cross-validations with a 90%-10% training-test split.

VI. Conclusions

The implementation of the GSC Algorithm and kNN Classifier with Euclidian Distance shows an improvement for the readings of damaged or incomplete characters using optical character recognition from a previous detectability strategy (Pattern Matching) from 95% to 96.76% ±0.405. Future work will consider increasing the dataset samples and the use of other distance metrics, as well as other classification algorithms.

References

[1] S. Mori, C. Suen and K. Yamamoto, “Historical review of OCR research and development”, *Proceedings of the IEEE*, vol. 80, no. 7, pp. 1029-1058, 1992.

[2] R. Casey and E. Lecolinet, “A survey of methods and strategies in character segmentation”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 7, pp. 690-706, 1996.

[3] Favata, J., “A Multiple Feature/Resolution Approach to Handprinted Digit and Character Recognition”, *Proceedings of the International Journal of Imaging Systems and Technology*, Vol. 7, pp. 304 - 311, 1996.

[4] D. Desrochers, Z. Qu and A. Saengdeejing, “OCR readability study and algorithms for testing partially damaged characters”, *Proceedings of 2001 International Symposium on Intelligent Multimedia, Video and Speech Processing. ISIMP 2001 (IEEE Cat. No.01EX489)*, 2001.

[5] R.O. Duda, P.E. Hart, and D.G. Stork. *Pattern Classification*. John Wiley & Sons, New York, 2001.

[6] Ballerini, L., Fisher, R., Aldridge, B., Rees, J., “Non-Melanoma Skin Lesion Classification Using Colour Image Data in Hierarchical KNNClassifier”, *Proceedings of 2012 IEEE International Symposium on Biomedical Imaging*, 2012.

[7] C. Rodriguez, “An Incremental and hierarchical KNNclassifier for Handwritten Characters”, *Proceedings of 2002 IEEE International Conference on Pattern Recognition*, vol. 3, pp 98-101, 2002.

[8] J. Hu, C. Yan, “Predicting Protein Subcelluar Localizations Using Weighted Euclidian Distance”, *Proceedings of 2007 IEEE 7th International Symposium on BioInformatics and BioEngineering*, pp 1370-1373, 2007.

[9] X. Qing-Song, L. Yi-Zeng, “Monte Carlo cross validation”, *Chemometrics and Intelligent Laboratory Systems*, Volume 56, Issue 1, 16 April 2001, Pages 1-11.