BLOB DETECTION TECHNIQUE USING IMAGE PROCESSING FOR IDENTIFICATION OF MACHINE PRINTED CHARACTERS

Aniket Patil
M.E Student,
Department of Electronics and Telecommunication,
Modern Education Society's College of Engineering, Pune, India.

Mrinai Dhanvijay
Assistant Professor,
Department of Electronics and Telecommunication,
Modern Education Society's College of Engineering, Pune, India.

ABSTRACT

Optical character recognition systems have been effectively developed for the recognition of printed characters. Optical character recognition is an awesome computer vision technique with various applications ranging from saving real time scripts digitally and deriving context based intelligence using natural language processing from the texts. One such application is the recognition of machine printed characters. This paper illustrates the technique to identify machine printed characters using Blob detection method and Image processing. In many cases of such machine printed characters there is similarity between character colour and background colour. There is mix up of reflected light and scattered light. Colour is not consistent across character area or background area. Paper explains how Blob detection technique is used for recognition of these machines printed characters.

KEYWORDS: OCR, Machine printed characters, Tesseract, OCV algorithm, OCR algorithm, Blob detection technique, Camera program for a Blister pack application.

INTRODUCTION

Optical character recognition is an awesome computer vision technique with various applications ranging from saving real time scripts digitally and deriving context based intelligence using natural language processing from the texts. One such application is the recognition of machine printed characters. This low cost Computer vision based technique can detect machine part features and serial numbers and creates one to one map between them for identification and quality control. The serial numbers are printed on metal parts. This required us to solve it in two stages. A preprocessing stage which extracts the region of text and segments out the characters and OCR stage which identifies the characters based on a pretrained model built using machine learning techniques [1]. OCR systems and research have continued to improve over the years and have now reached a point that some researchers deem the recognition of machine printed character images. Affordable commercial software packages are also available, with some advertised claiming recognition accuracy rates above 99%. Optical character recognition also called optical character reader is a system that provides a full alphanumeric recognition of printed or handwritten characters at electronic speed by simply scanning the form. More recently the term intelligent character recognition (ICR) has been used to describe the process of interpreting image data, in particular alphanumeric text. OCR performs different types of functions. Forms containing character images can be scanned through scanner and then recognition engine of the OCR system interpret the images and turn images of handwritten or printed characters into ASCII data (machine readable characters) [2].

Optical character recognition technology has been used extensively in commercial applications since 1970's, and is used today for automating tasks such as passport processing, secure document processing (checks, financial documents, bills), postal tracking, publishing, consumer good packaging (batch codes, lot codes, expiration dates) and clinical applications. This paper presents a real time engraved number recognition on machine parts.

MACHINE PRINTED CHARACTERS

Machine printed characters are usually a limitation for character recognition as their extraction is difficult due to small contrast with the background. In many cases colour is not consistent across character area or background area. The images of some machine printed characters are shown below.

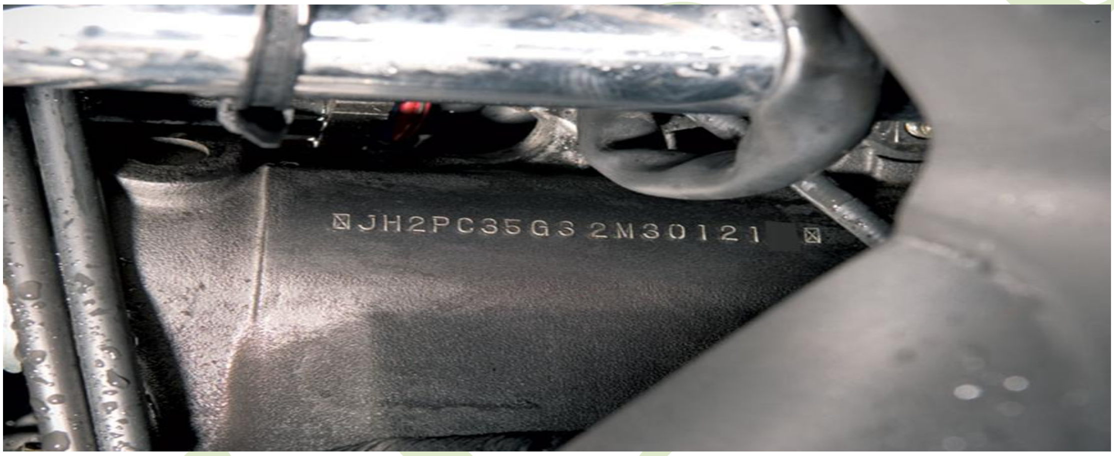


Fig.1 Characters printed on Industrial Machine

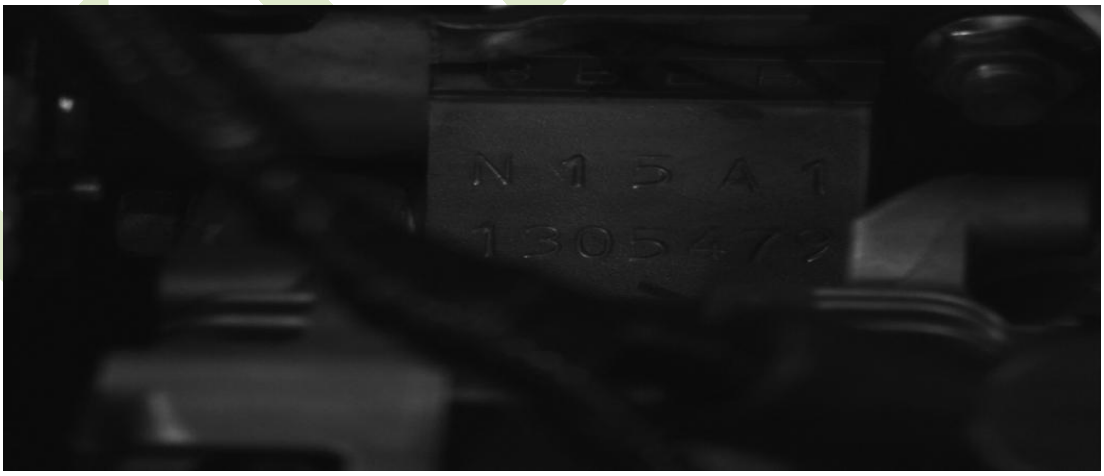


Fig.2 Characters printed on a Machine

IMPLEMENTATION DETAILS

Configuration of Global Shutter Camera:

We are using global shutter camera to capture the images on machines. Global shutter camera provides up to 5MP resolution with CCD sensors. Camera provides faster speeds without higher costs. It provides maximum frame rate up to 50 fps for 1ms exposure. This camera offers excellent image quality and a flexible high performance interface. We are using agile approach. We are implementing this system on real time. The Tesseract OCR software is used to test our results.

Multiple Approaches to Character Classification:

This OCR technology explores new spaces for character representation. OCR provides new approaches to character classification in different spaces.

Libraries:

Open CV library we are using in this system. Linux operating system along with Python programming is used to implement our system.

TESSERACT

Tesseract is open source OCR engine. It was firstly developed between 1984 to 1994 at HP. This open source OCR engine works with independently developed page layout analysis technology. Tesseract accepts input image as a binary image. Handle both, the traditional black on white text and also inverse-white on black text.

ARCHITECTURE OF TESSERACT

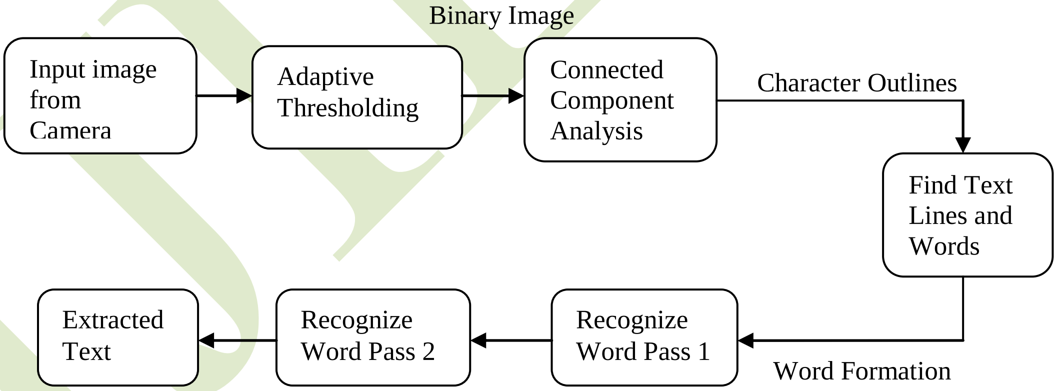


Fig.3 Architecture of Tesseract

Input image of character string is applied to adaptive thresholding. Due to adaptive thresholding input grey or colour image is converted into binary image. This binary image is applied to connected component analysis to form character outlines. Nesting of character outlines is done which gathers the outlines together to form a Blob. Such blobs are organized into text lines. Tesseract performs activity to recognize words. This recognition activity is mainly consists of two passes. The first pass tries to recognize the words then satisfactory word is passed to adaptive classifier as training data, which recognizes the text more accurately. During second pass, the words which were not recognized well in the first pass are recognized again through run over the page.

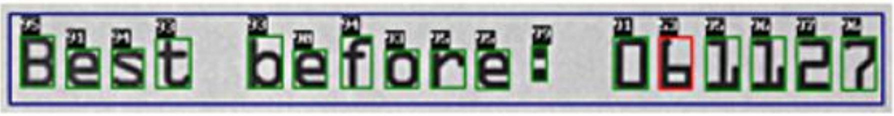
ALGORITHMS

Optical Character Verification (OCV) Algorithm

Optical character verification, or OCV, is an algorithm that verifies a taught-in text string. The OCV function gives the result true (if the correct string of characters found) or false (if the incorrect string of characters found). For identification of machine printed characters OCV algorithm gives up to 85% accuracy. But the drawback of OCV algorithms is, it is not useful for edge detection of machine printed characters.

Example: OCV

OCV: Teach String.



OCV: Recognize string. A misprinted or incomplete character is identified as a fail.

Optical Character Recognition (OCR) algorithm

Optical character recognition, or OCR, is an algorithm that reads or recognizes unknown text, where each letter is compared with a taught-in font. The OCR function gives the results as stated below:

1. The read string, i.e. the sequence of characters.
2. True or false, i.e. if the reading of characters is successful or if one or more characters are not recognized as part of the font.

Optical character recognition algorithm gives character identification accuracy up to 90%. But OCR algorithm encounters a problem when font size of characters changes. OCR algorithm faces difficulty in reading similar characters like I and i, O and 0. Tight distance between characters also pose difficulties to identify machine printed characters. Also edge detection is not done using OCR algorithm. To overcome all these difficulties Blob detection technique is introduced.

BLOB'S DETECTION TECHNIQUE

A Blob analysis algorithm is used to find and count objects, and to make basic measurement of their characteristics [4]. The purpose of analysis is to determine whether the results obtained from an operation are accurate, logical and true. Image processing software comprises complex algorithms that have pixel values as inputs. Today's image analysis software packages include both old and new technologies. Most significant is the relationship between the old blob analysis method and the new age-detection technique. For image processing, a blob is defined as a region of connected pixels. Blob analysis is the identification and study of these regions in an image [5] [6]. The algorithm discerns pixels by their value and places them in one of two categories. Viz. foreground (typically a pixel has a non-zero value) or the background (pixels with a zero value).

IJIERT

Since a blob is a region of touching pixels, analysis tools typically consider touching foreground pixels to be part of the same blob. Consequently what is easily identifiable by the human eye as several distinct but touching blobs may be interpreted by software as a single blob. Furthermore, any part of a blob that is in the background pixel state because of lighting or reflection is considered as background during analysis [7].

The performance of a blob analysis operation depends on a successful segmentation of the image that is, separating the good blobs from the background and each other as well as eliminating everything else in the image that is not of interest. Segmentation involves a binarization operation [11]. If simple segmentation is not possible due to poor lighting or blobs with the same grey level as parts of the background, we must develop a segmentation algorithm appropriate to a particular image.

The acquired image may contain noise or spurious blobs or holes that may be caused by noise or lighting. Such extraneous blobs can interfere with the blob analysis results [12]. If the image contains several extraneous blobs, we should pre-process the image before using it. Preprocessing refers to any steps made to clean up the image before analysis and can include thresholding or filtering.

Example:

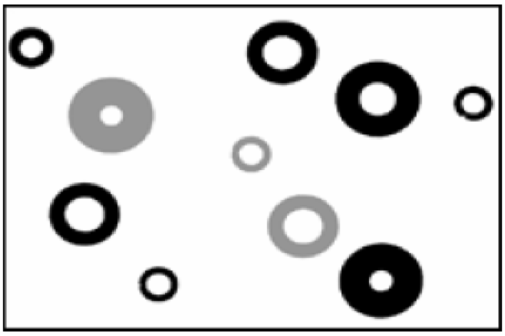


Fig.4

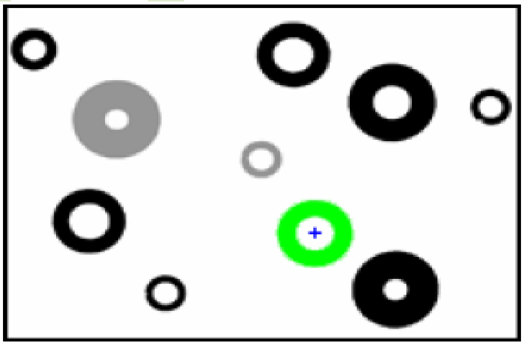


Fig.5

Blobs of four different sizes and two grey levels are shown in figure.4 whereas Blob found by double search criteria using grey scale and area thresholding are shown in figure.5

CAMERA PROGRAMMING

A program can branch to do different things depending on intermediate or final results. This is obtained through conditional instructions, for example the IF statement that is used for pass/fail treatment. Calculations and string (text) operations are handled in the program by expression evaluations. An expression is a written formula that can contain numbers, variable references, or strings. Depending on the type of expression, the evaluation gives different results as follow.

1. Boolean: 1 or 0, true or false, good or bad.
2. Numeric: a number, for example 3.1415
3. String: text, for example “Best before May 2010” or “Lot code AAA”.

A common situation is when a part of the program needs to be reused frequently. Instead of copying the code over and over again, it can be packaged in a macro that can be exported to other applications.

Processing and Analysis:

Collection of all Blobs together forms a blister. This blister consists of pills. A blister pack needs inspection before the metal foil is glued on to seal the pack. The task is to inspect each blister for pill presence and correctness. If any of the blisters is faulty, camera conclude a fail and the result is communicated to reject the blister pack [9]. A backlight is used to the pill contours. This application is solved by our Blob analysis technique depending on accuracy needs and cycle time requirements. Pills bounded by green rectangle shows correct pills whereas pills bounded by red rectangle shows incorrect pills in a blister pack. A blister pack with pills is shown in figure below.

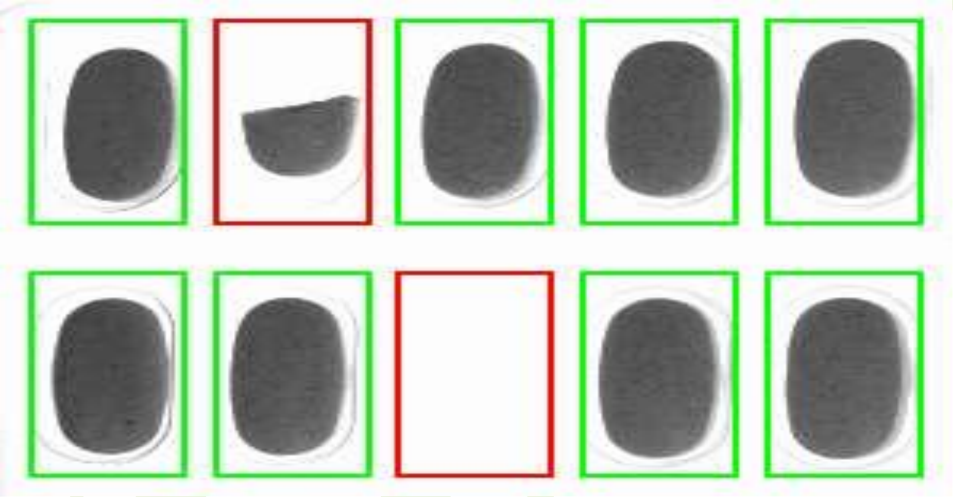


Fig.6 Blister pack with pills

Flow Chart for a blister pack application

Camera imaging technology is used to detect defects in indivisual blister. Conditional instruction IF-ELSE is used in the flow diagram for a blister pack application. If the number of Blobs match with the previously counted blobs then we call it as Pass statement and output is set to zero. Conversely if the number of blobs are not matched with the counted blobs then we call it as fail statement and output is set to one [10]. These incorrect blobs are again send back to capture the image once again until the blobs of correct size matches with number of correct blobs.

Blob detection technique gives edge detection of characters and helps to remove unwanted area from character string. Blob detection technique is not affected by font size. Characters printed on machines with any font size and of any font format can be identified using blob detection technique with almost 95% accuracy.

Flow chart for a blister pack application is shown below.

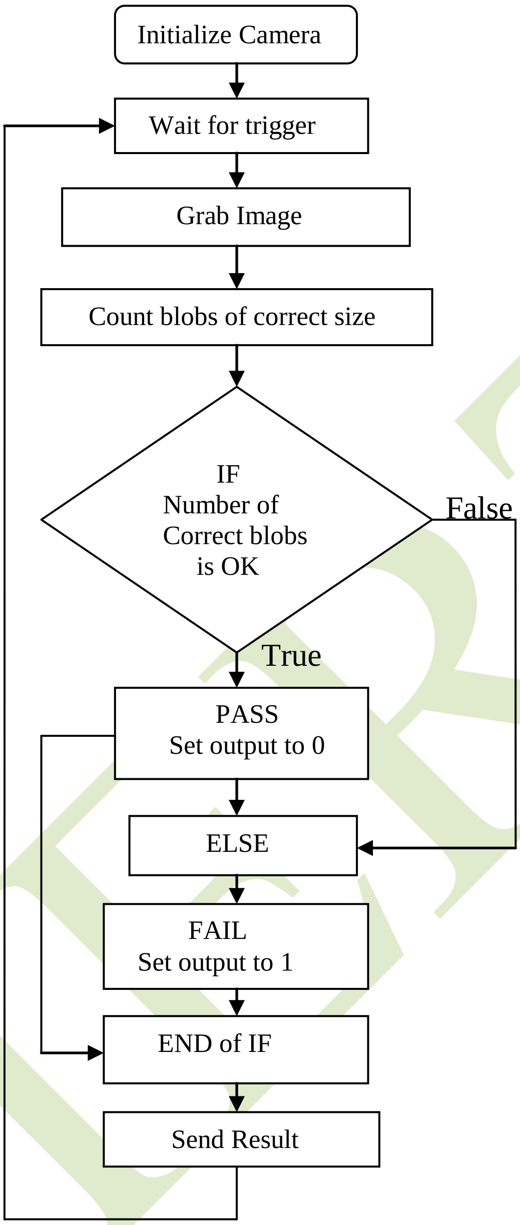


Fig.7 Flow chart for a blister pack application

RESULTS

Results are shown in tabular form as shown below.

Table1. Comparison of results

Sr. No	Algorithm	Character Identification	Edge Detection	Accuracy
1.	Optical Character Verification (OCV)	Yes	No	85%
2.	Optical Character Recognition (OCR)	Yes	No	90%
3.	Blob Detection	Yes	Yes	95%

Compare with existing algorithms like Optical character recognition (OCR) algorithm and optical character verification (OCV), blob analysis gives more accurate results up to 95% for machine printed character recognition and edge detection. The application is implemented in Python using open CV library in Linux environment with a single camera view.

CONCLUSION

Thus Blob detection technique is used for identification of machine printed characters. Edge detection of characters is also achieved with removal of unwanted blobs in an image. Blob analysis is just one example of the image processing tools available today and is used widely for identification of machine printed characters.

ACKNOWLEDGEMENT

I would like to thanks to my respected guide Prof. Mrinai Dhanvijay, Assistant Professor in Electronics and Telecommunication Department, MESCOE, Pune, Maharashtra, India. I would like to thank her for continuous encouragement throughout this work.

REFERENCES

[1] S. Naz, K. Hayat. *"Optical character recognition of Urdu-like cursive scripts"*, ELSEVIER. Pattern Recognition, 47, 2014, pp.1229-1248.
[2] R. Minetto, N. Thome, M. Cord. *"A text detection system for automatic indexing of urban scenes"*. Computer Vision and Image Understanding, 122, 2014, pp.92-104.
[3] F. Caillette and T. Howard, *"Real time marker less human body tracking using colored voxels and 3-D blobs"*. In Proc. ISMAR, 266-267, 2004.
[4] C. Damerval and S. Meignen, "Blob detection with wavelet maxima lines". IEEE Signal Processing Letters, 14(1):39-42, 2007.
[5] P. E. Forssen and G. Granlund. "*Robust multiscale extraction of blob features"* In proceedings of the 13th Scandinavian Conference on Image Analysis, LNCS 2749, 11-18, 2003.
[6] G. Gerig, G. Szekely, G. Israel and M. Berger. Detection and characterization of unsharp blobs by curve evolution. Information Processing in Medical Imaging, 165-176, 1995.
[7] S. Hinz, "Fast and Subpixel Precise Blob detection and Attribution" in Proceedings ICIP'05, Genua. pp.457-460, 2005.
[8] S. Lu, G. Tsechpenakis, D. N. Metaxas, M. L. Jensen, and J. Kruse. "*Blob analysis of the Head and Hands: A method of Deception Detection"*, Hawaii International Conference on System Science (HICSS'05), 2005.
[9] G. Li, T. Liu, J. Nie, L. Guo. *"Detection of blob objects in microscopic zebrafish images based on gradient vector diffusion"*. Cytometry A, 71:835-845, 2007.
[10] Wang, F.; Ren, X.; Liu, Z. *"A Robust blob recognition and tracking method in vision based multi-touch technique"*. In Proceedings of International Symposium on Parallel and Distributed Processing Applications (ISPA'08), Sydney, Australia 10-12 October 2008; pp. 971-974.
[11] A. Ming, H. Ma. *"Blob detection in color images"*. Proc. of the 6th ACM CIVR, 2007, PP.364-370.
[12] N. khanina, E. Semeikina, D. Yurin. Color blob and Ridge detection. Proc. of 10th conf. on *"Pattern Recognition and Image Analysis: New Inform. Technologies*", 2010, V.1.PP.289-292.