

Counting the Number of Objects of An Image Using Image Processing Algorithms Based on Open CV

D Bhanu Prakash¹ and G Ravi Kumar Reddy²

¹CVR College of Engineering, Department of ECE, Hyderabad, INDIA
Email: pbhanududi@gmail.com

²CVR College of Engineering, Department of ECE, Hyderabad, INDIA
Email: ravigrkr@gmail.com

Abstract –The number of objects counting is the estimation of objects in an image or video frame. It is a big challenging task in image processing. Some of main applications are counting blood cells in medical images, counting copper cores in wire, number of logs in timber truck, and monitoring crowds in meetings etc. This work presents a method of image edge detection with contour tracking to determine the exact number of objects in an image based on open computer vision (OpenCV) library. The experimental results show that OpenCV based algorithms are simple, with high code integration and accuracy.

Keywords: Computer vision, Contour, Image, Video, Medical image, Copper core, Logs.

I. OPEN CV OVERVIEW

Open computer vision (OpenCV) is an Intel open source computer vision library. The library has more than 2500 image processing algorithms is written in C and C++ and runs under Linux, Windows and Mac operating systems. It has C++, python, Mat Lab interfaces. OpenCV library can be used to build real time image processing with high computational efficiency [1].

A) History

OpenCV was introduced by Intel in 1999 by Gary Bradsky and Vadim Pisarsky. First alpha release came out in 2000 at the IEEE Conference on computer vision and pattern recognition. In mid-2006 there is an active development in OpenCV by “Willow Garage” team. A version 1.1 “pre-release” was released in October 2008. Now OpenCV getting support from a non-profit foundation called “OpenCV.org”.

B) Open CV Modules

Open CV is classified into five main modules [2]. i.e.

- a) The CV Module
- b) The MLL Module
- c) High GUI Module
- d) The CX core Module
- e) CV Aux Module

Figure 1 shows the basic structure of OpenCV. The CV module has the basic image processing and computer vision algorithms. The ML is the machine learning library which includes many classifiers and clustering tools. The

High GUI has algorithms to read/save images and videos. The CX core contains the basic data structures and contents. The CVAux has algorithms for face recognition, stereo vision, texture descriptor etc. [3].

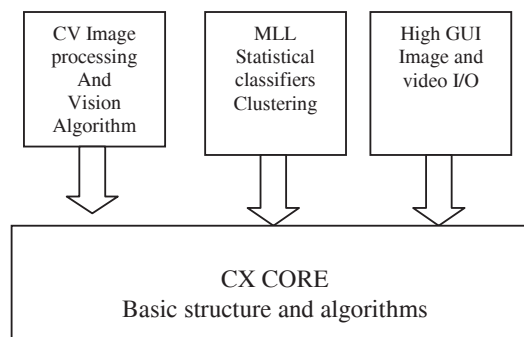


Figure (1): Basic structure of Open CV

C) Portability

OpenCV algorithms can run in any system that has C and C++ platform. This property makes OpenCV portable.

D) Image structures in OpenCV

Image in digital form is stored as an array of numerical information. The basic image structure hierarchies are of three types.

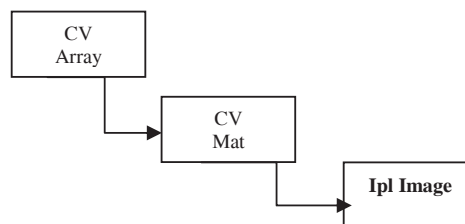


Figure (2): Basic data Structure in Open CV

OpenCV has a built-in data type to represent image matrices called Mat. For example, to initialize an image of dimensions 300x500 pixels, the syntax is:

Mat image (300,500)

Image Color Storing Methods: These methods are classified into three types.

Binary:

It is the most fundamental way to store an image. The image is least defined black and white in color. Intensity information is stored either as 0 or 1.

Example: Mat image (300,500)

Gray Scale:

Basically black and white image, but more defined than a binary image. In this, the intensity value stored usually ranges from 0 to 255, i.e. 1 byte of data.

Example: Mat image (300, 500, CV_8U1)

RGB Image:

It is one of the most standard methods to store color images. It is a multichannel method, i.e. there are three different intensities stored for each pixel. The three channels correspond to intensities of Red, Blue and Green color.

Example: Mat image (300, 500, CV_8U3)

Here, pixel intensity value is represented in CV_8U3, 8 means 8 bits; U means unsigned char and 3 means 3 channels.

Ipl Image structure: Structure from Intel Image Processing Library to represent the image data.

Figure 3 shows the source image captured by camera.



Figure (3): The source image of copper core wire

II. COLOR TO GRAY CONVERSION

Source images are obtained generally in color. The RGB color images contain three components for each pixel. The colors recorded by a camera are heavily dependent on the lighting conditions. They require lot of processing. For facilitating processing, color image will be usually converted to gray scale [4]. A gray scale image is an image in which the value of each pixel is an 8 bit value (0-255 range).

It carries only intensity information. The function *cvtColor* is used to convert RGB image into gray image.

Example: cv.CvtColor (src, dst, code);

In above function **src** is input image, **dst** is output image and **code** is color space conversion code like *cv_RGB2GRAY*.

The effect of RGB to Gray scale image is well demonstrated in the Figure 4

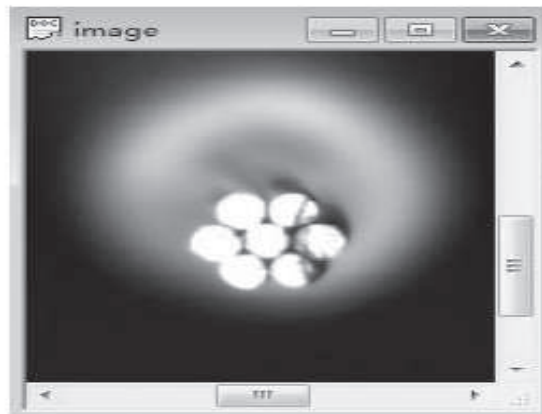


Figure (4): RGB to Gray scale image

While capturing the image, some noise is introduced to source image due to camera sensor. Image sensor working is affected by temperature and light levels are major factors for producing the noise. The noise in image decreases detection accuracy. It is necessary to remove the noise in captured image. There are many methods available for removing the noise in images. Most frequently used method is smoothing. This method is also called blurring. Five smoothing functions are available in openCV which can be called by CVSmooth function.

The available functions are

- CV_BLUR,
- CV_BLUR_NO_SCALE,
- CV_MEDIAN,
- CV_GAUSSIAN and
- CV_BILATERAL

Median filtering method is kind of nonlinear smoothing technique and the gray value of its every pixel is set to a middle value of all the gray value of the pixel in a neighboring window [5].

Example: CVSmooth (image, blurring, CV_BLUR, 7, 7);

CVSmooth (image, medimg, CV_MEDIAN, 7, 7);



Figure (5): Image after smooth processing

IV. THRESHOLDING

The segmentation is used to divide the image into separate regions containing pixels with similar properties. This separation is made on the basis of intensities between object pixel and background pixels [6].

The Image segmentation is done by thresholding. It is a nonlinear method that converts a gray scale image into binary image. Pixel values are assigned to one value (may be white), if value is greater than threshold value; else, it is assigned to another value (may be black) if value is less than threshold value [7]. The function for thresholding is *CV.Threshold*.

OpenCV has different types of thresholding. They are *CV.THRESH_BINARY*, *CV.THRESH_BINARY_INV*, *CV.THRESH_TRUNC*, *CV.THRESH_TOZERO*, *CV.THRESH_TOZERO_INV*.

In the present work, for a micro wire picture, the object can be seen as the copper core, and the other is the background. The results of thresholding can retain structural characteristics of copper core information. For the purpose of separating copper core from background as much as possible, threshold function is to be used to segment cores in a wire. Figure 6 shows the resultant image after thresholding.



Figure (6): Image after Thresholding

V. MORPHOLOGICAL OPERATIONS

Morphological operations process the images according to shapes. They apply a defined structuring element to an image, obtaining a new image where the pixels at positions (x_i, y_i) are computed by comparing the input pixel values at positions (x_i, y_i) and their neighborhoods. Depending on the structuring element selected, a morphological operation is more sensitive to one specific shape or the other. The two basic morphological operations are the dilation and the erosion [8]. Dilation adds pixels from background to the boundaries of the objects in an image, while the erosion removes pixels. Here is where the structuring element is taken into account to select the pixels that are to be added or deleted. In the dilation the value of the output pixel is the maximum of all the pixels in the neighborhood. Using the erosion, the value of the output pixel is the minimum value of all the pixels in the neighborhood. The other image processing operations can be defined by combining

the dilation and the erosion, such as the opening and closing operations, and the morphological gradient. The opening operation is defined as erosion followed by dilation, while closing is its reverse operation –dilation followed by erosion. Therefore opening removes small objects from an image while preserving the larger ones and closing is used to remove small holes while preserving the larger ones in a manner similar to opening. The morphological gradient is defined between the dilation and the erosion of an image.



Figure (7): The morphological processing image

VI. CONTOUR TRACKING

Contours are defined as a curve joining all the continuous points along the boundary, having same color or intensity. Contour tracking extracts the boundary of objects in an image. It is also used to extract the shapes of objects. For better accuracy, use binary images. So before finding contours apply threshold or canny edge detection for getting binary images [9].

In OpenCV, to find white objects from black background, we use *findContour* function

Example: *CV.FindContour(src, contours, hierarchy, CV_RETR_tree, CV_CHAIN_APPROX_NONE);*

By using this method, it clearly distinguishes the boundary of the objects, so it determines the number of objects accurately. The effect of contour tracking is shown in figure 8. From which we can clearly distinguish the exact number of the core in the wire.



Figure (8): The effect of contour tracking

VII. RESULTS AND DISCUSSION

An attempt is made to count the number of objects in images. The segmentation results shows that, some images segmented more than actual objects and some images segmented less than actual objects. The average objects segmentation gives around 97%, where individual objects segmentation varies between 92% and 102%. However, variation in the results of individual objects are did not matter much as the aim of this paper is to calculate the average number of objects in an image. The increase or decrease of few objects does not make any difference as results are very much closer to actual average.

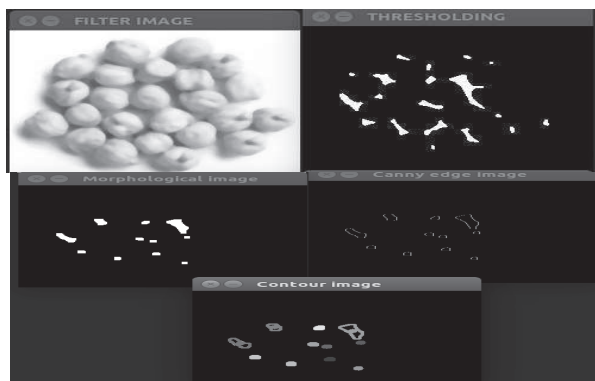


Figure (9): Results of image processing technique to count number of seeds

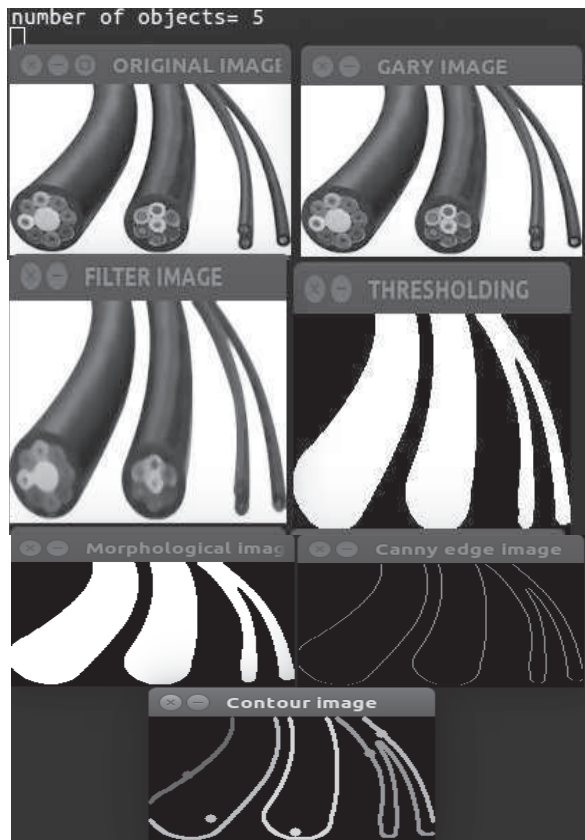


Figure (10): Results of image processing Technique to count number of cables

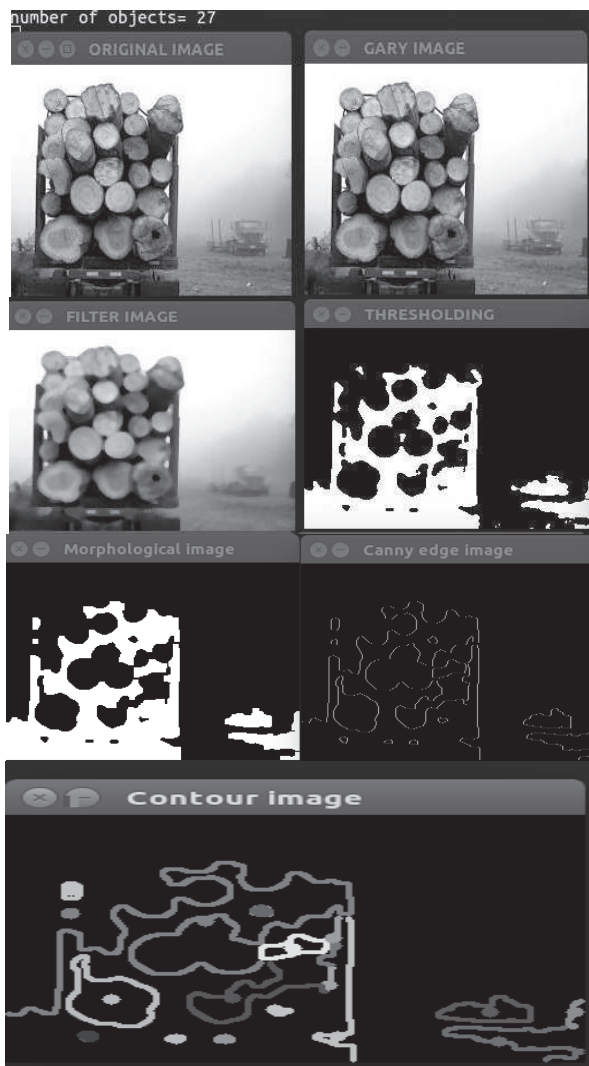
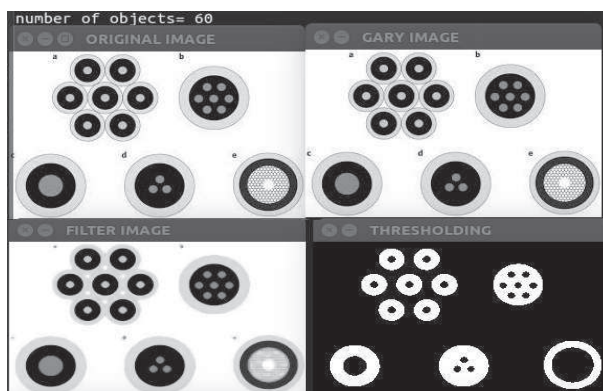


Figure (11): Results of image processing technique to count number of logs



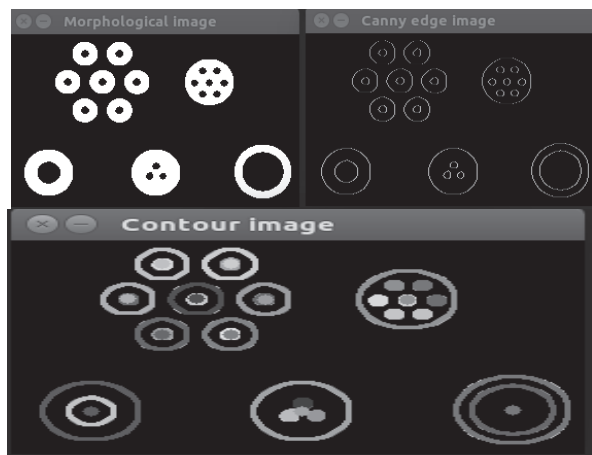


Figure (12): Results of image processing technique to count number of cores in cables.

- [9] M. W. Ren, J. Y. Yang, and H. Sun, "Tracing boundary contours in a binary image[j]," *Image and Vision Computing*, vol. 20, pp125-131, 2002

VII. CONCLUSION

Computer vision or machine vision is the automated extraction of information from images and video frames. This technology has been gradually adopted by industry to monitor the product specifications to improve test efficiency and accuracy. With the help of OpenCV and image processing algorithms, first we preprocess the target image, secondly to make clearly distinguish between boundary of objects and its background, we used morphological methods. Finally contour tracking is used to count exact number of objects.

REFERENCES

- [1] G. Sezer, A. Ertüzün, A. Erçil, "Independent component analysis for texture defect detection," in *Proc. 6th German-Russian Workshop OGRW-6-2003*, Novosibirsk, Nov. 2003, pp. 210-213
- [2] A. Kumar, "Computer-vision-based fabric defect detection: a survey," *Industrial Electronics, IEEE Transactions on*, vol. 55, pp. 348–363, Jan. 2008.
- [3] C. Beirão, M. Figueiredo, "Defect detection in textile images using Gabor filters," in *Proc. ICIAR'2004, LNCS, Springer Verlag*, vol. 3212, pp. 841-848, 2004.
- [4] A. Koschan, "A Comparative Study on Color Edge Detection", in *IEEE Proc. ACCV'95*, Singapore, 1995, pp 478-545,
- [5] R. S. Deepthi and S. Snakaraiah, "Implementation of mobile platform using Qt and OpenCV for image processing applications," in *2011 IEEE Conference on Open Systems*, 2011, pp 290-295,
- [6] Mello, A. B. Carlos, Costa, H. M. Antonio, "Image thresholding of historical documents using entropy and ROC curves," *Lecture Notes in Computer Science*, vol. 3773, pp 905-916, 2005.
- [7] X. J. Jia, "Fabric defect detection based on open source computer vision library OpenCV," *2010 2nd International Conference on Signal Processing Systems, Dalian, 5-7 July 2010*, pp.
- [8] F. Wang, G. T. Jiao, and Y. Du, "Method of fabric defect detect in based on mathematical morphology," *Journal of test and measurement technology*, vol. 21, pp. 515-518, 2007.