International Journals of Advanced Research in Computer Science and Software Engineering
ISSN: 2277-128X (Volume-8, Issue-8)

Research Article | **August 2018**

An Efficient Algorithm for Brick Counting

Jasleen Kaur
Research Scholar
jasleen.7956@gmail.com

Sanpreet Singh
Research Scholar
sany26.com@gmail.com

Satinderpal Singh
Research Scholar
satinder.goraya91@gmail.com

Abstract: ***Counting is an important task for the Quantitative Analysis of things. Construction progress demands regular checking of bricks to get the estimation of number of bricks ordered, used or trashed; to prepare a report of construction progress. Manual counting always seems to be subjective judgement of count. It doesn't assure accuracy, as it may be conceivable that same object gets numbered twice or thrice and sometimes, it got skipped by mistake which in turn leads to errorneous results. It always demands continous attention, eye fatigue and a good amount of time. To overcome the issues of manual counting, one needs an automated system. This paper aims at providing objective judgement to count the bricks. It paper presents an approach that will count the bricks installed on a wall accurately and in a very short time.***

Keywords: Brick count, RGB to HSV, Otsu Threshholding, Morphological Operation

I. INTRODUCTION

Object counting is a common trend nowadays. The need of Counting the objects arises in day to day life with aim to provide an easy method which will span less time and yields accuracy in results. To overcome this need, many research work has been done that provides automated method to resolve the issue of counting objects. Challenge of automated system is to identify the object from its background. Problem seems to be in cases where the objects are not much discernible from background. In the current work, an algorithm to count the bricks from brick wall is presented. The approach used in this paper provides reliable and thorough tested results. An open-source API known as OpenCV-Python is used and numpy, matplotlib libraries are additionally utilised.

II. LITERATURE

Youngeun An, Muhammad Riaz and Jongan Park[5] in the paper "CBIR Based on Adaptive Segmentation of HSV Color Space" proposed an algorithm based HSV color space. The local Histogram is used to extracting color information from an image, but its drawback is that other information like shape, position, orientations will be discarded. HSV Color space gave the color distribution in the image that helps in getting a good histogram result.

Victor Lempitsky, Andrew Zisserman[2] in the paper "Learning To Count Objects in Images" proposed a new supervised learning framework in which continuous density function whose integral over any image region gives the count of objects within that region. The hard task of learning to detect and localize individual object instances was to find continuous density function which will give the detection of an instance in a image and its integration over whole image is taken to count the total number of objects. One simply needs to extract feature vectors at each pixel and multiply them with the learned linear coefficients to obtain a density value. The density can then be integrated over an arbitrary shaped region of interest (e.g. the full image) to estimate the number of objects present in it. The proposed system is flexible enough as it can accept any domain specific visual features. The system gives good accuracy in counting objects for applications involving real time processing.

L. Hui1 and I. Brilakis[1], in the paper "Real-Time Bricks Counting for Construction Progress Monitoring" proposed an automatic counting algorithm to count number of bricks from an image. In this paper, color thresholding is used to extract color properties of bricks. Laplacian of Gaussian is used that provides the edges of bricks. The edge maps obtained using LOG is then compared with bricks properties, its shape, orientations, size. The experimented results give the precision of 98%, indicating that the algorithm can detect the bricks accurately and recall of 90%, indicating that the algorithm is able to detect most of the bricks. But this algorithm is restricted to bricks detected using Red Color, Rectangular Shape, Size as parameters.

III. METHOD

For the research work to perform, a new collection of sample images is taken using camera. The images used for the research work are taken in natural day light, with some cracks on bricks. The processing of image starts with

denoising the captured image. To perform denoising, Gaussian filter is preffered. RGB Image is converted to HSV color space to perform good segmentation results. In HSV, color extraction is more robust[4]. Thresholding is performed on HSV image after converting it to grayscale to yeild a binary image with bricks in white and background in black. After performing morphology correction, much of its noise was removed while performing morphological opening and filling of bricks was done at accurate level. Now there was existence of only some small size connected regions which get rescue during morphological opening and are observed as noise. Statistics analysis is observed for small connected regions. A particle size of 30 pixels is observed and consider as a noise. Therefore filtering of remaining noise is performed in such a manner so that connected pixels with area 30 or less would be considered as noise. The objects remained after filtering out noise from image can now be consider as a desired objects i.e the bricks in the image. Contour finding is performed to find the contour of bricks and access its characteristic. Bricks are counted using counter that iterates over every contour in an image.

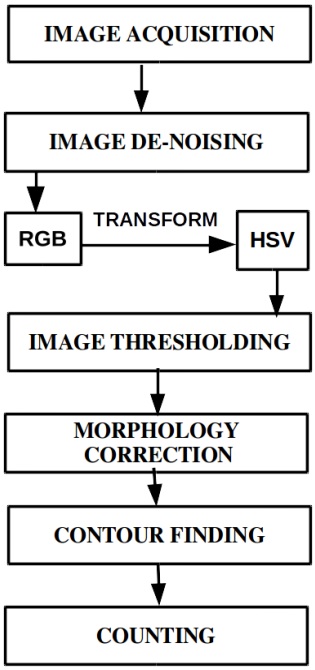


Figure 1. WorkFlow of Brick Counting Algorithm

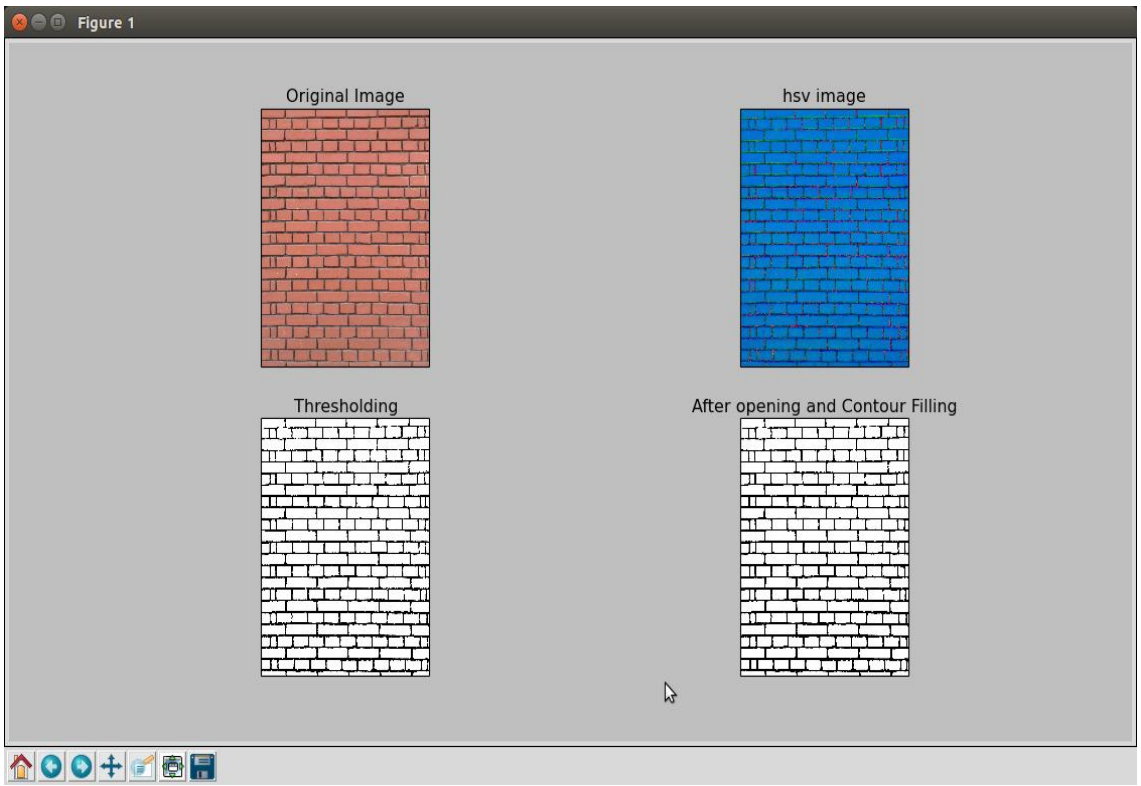


Figure 2. Original image, hsv, thresholded, morphological opening and contour filing image

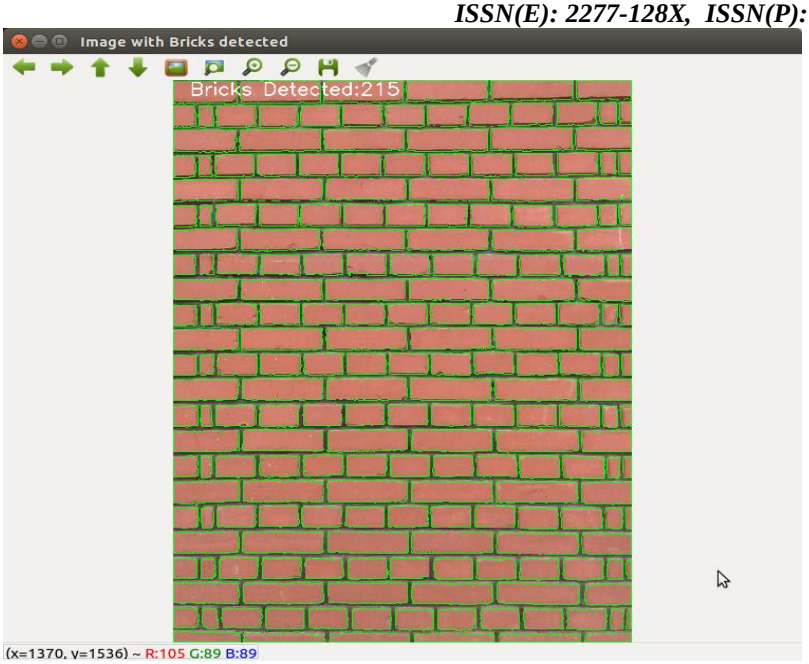


Figure 3. Output image with the count of bricks.

Image segmentation using HSV Colorspace: In HSV colorspace, Value channel gets isolated from Hue and Saturation channel and a histogram construction or thresholding can be done using only Saturation and Hue. That is why for an image segmentation based on color extraction, it is preferable to convert RGB to HSV to get a more improved results during thresholding an image into binary[2]. The HSV color space is in a general sense unique in relation to the broadly known RGB shading space since it isolates out the Intensity (luminance) from the shading data (chromaticity). The HSV can determine the intensity and shade variations near the edges of an object, thereby sharpening the boundaries and retaining the color information of each pixel. The HSV features helps the identity of the colors even at different intensity levels. This makes the HSV-based features very useful in running segmentation algorithms like clustering on the approximated pixels[2].

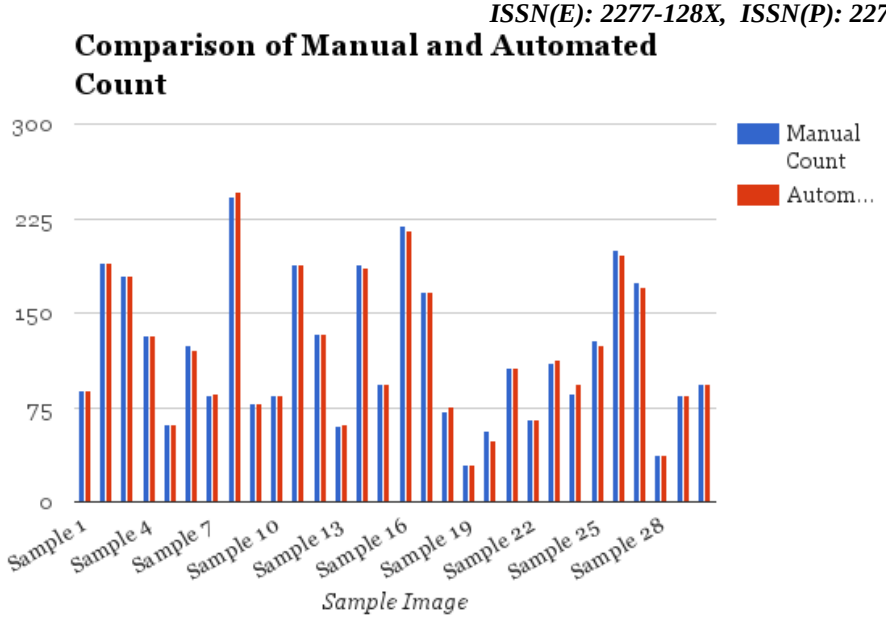
IV. RESULTS

TP = TRUE POSITIVE
FP = FALSE POSITIVE
FN = FALSE NEGATIVE

Comparison Table of Manual counting and Automated Counting

Sample Image (.jpg)	Manual Count	Automated Count	Precision (tp/tp+fp)	Recall (tp/tp+fn)	Time Taken(seconds)	Percentage Error
Sample 1	89	89	1	1	0.36	0
Sample 2	190	190	1	1	2.04	0
Sample 3	180	180	1	1	0.18	0
Sample 4	132	132	1	1	2.05	0
Sample 5	61	61	1	1	0.18	0
Sample 6	124	121	1	0.9758064516	1.95	-3.370786517
Sample 7	85	86	0.988372093	1	1.21	1.123595506
Sample 8	242	246	0.9837398374	1	5	4.494382022
Sample 9	78	78	1	1	0.96	0
Sample 10	84	85	0.9882352941	1	1.22	1.123595506
Sample 11	189	189	1	1	2.03	0
Sample 12	133	133	1	1	0.15	0
Sample 13	60	62	0.9677419355	1	0.97	2.247191011
Sample 14	188	186	1	0.9893617021	3.99	-2.247191011
Sample 15	94	93	1	0.9893617021	1.36	-1.123595506
Sample 16	219	215	1	0.9817351598	2.71	-4.494382022
Sample 17	167	167	1	1	1.73	0
Sample 18	72	76	0.9473684211	1	1.73	4.494382022
Sample 19	29	29	1	1	0.37	0
Sample 20	57	49	1	0.8596491228	0.96	-8.988764045
Sample 21	107	107	1	1	1.95	0
Sample 22	66	66	1	1	1.09	0
Sample 23	110	113	0.9734513274	1	2.6	3.370786517
Sample 24	86	93	0.9247311828	1	1.8	7.865168539
Sample 25	128	125	1	0.9765625	3.87	-3.370786517
Sample 26	200	196	1	0.98	7.6	-4.494382022
Sample 27	175	170	1	0.9714285714	7.6	-5.617977528
Sample 28	37	37	1	1	0.48	0
Sample 29	84	84	1	1	0.81	0
Sample 30	94	94	1	1	1.44	0

Average Precision	Average Recall	Average Time Taken	Average percentage Error
0.9924546697	0.9907968403	2.013	0.2996254682



It has been observed that performed research work on dataset of 30 images yield results as:

Average Precision = 99.24%

Average Recall = 99.07%

Average Percentage error = 2.9%

The average time taken by proposed scheme is 2.03 seconds to process an image of approx. 1200*1800 size

V. CONCLUSION

The results proves that the paper presents an efficient approch which gives the good estimation of count of laid bricks in wall with acceptable error, covering the minute cracks in bricks by itself. The work done is able to count the bricks with an accuracy of 97%, precision of 99.04% and recall of 99% is observed. The average time taken by proposed scheme is 2.03 seconds to process an image of approx. 1200*1800 size. Errorneous result is expected in case of bricks with much variation in illumination and the bricks which doesn't have distinct separation as if cement get spread over bricks while installation. This algorithm with further improvement can be used to find the size of bricks and can be used to count different objects.

REFRENCES

[1] L. Hui and I. Brilakis, "Real-Time Brick Counting for Construction Progress Monitoring", Computing in Civil Engineering, 2013.

[2] V. Lempitsky and A. Zisserman. Learning to Count Objects in Images. NIPS, 2010

[3] W. Hou, Z. Duan and X. Liu, 'A Template-Covering based algorithm to Count the Bundled Steel Bars', 2011 4th International Congress on Image and Signal Processing, 2011.

[4] W. Mustafa, H. Yazid and S. Yaacob, "Illumination correction of retinal images using superimpose low pass and Gaussian filtering", 2015 2nd International Conference on Biomedical Engineering (ICoBE), 2015.

[5] Y. An, M. Riaz and J. Park, "CBIR Based on Adaptive Segmentation of HSV Color Space",2010 12th International Conference on Computer Modelling and Simulation, 2010

[6] S. Sural, Gang Qian and S. Pramanik, "Segmentation and histogram generation using the HSV color space for image retrieval", Proceedings. International Conference on Image Processing.

[7] H. Jing, L. Peiyuan and C. Hanwei, 'Research on the Rice Counting Method Based on Connected Component Labeling', 2014 Sixth International Conference on Measuring Technology and Mechatronics Automation, 2014, Print ISBN: 978-1-4799-43-48

[8] Y. Toh, T. Ng and B. Liew, "Automated Fish Counting Using Image Processing", 2009 International Conference on Computational Intelligence and Software Engineering, 2009.

[9] A. Chavan, A. Shastri, R. Shastri and S. Deosarkar, "Counting of Frozen Semen Straws Using Image Processing", 2013 Third International Conference on Advances in Computing and Communications, 2013.

