(IJIASE) 2018, Vol. No. 4, Jan-Dec **e-ISSN: 2454-9258, p-ISSN: 2454-809X**

ANALYZING AND IDENTIFYING THE RAW RICE GRAIN SORTING TECHNIQUES EMPLOYING MATLAB TO DEVELOP AN EFFICIENT SORTING ALGORITHM FOR COMMERCIAL EXPLOITATION USING PROPRIETARY PROGRAMMING

Rahul Garg,
Bachelor of Engineering (B.E.), ECE)
Thapar Institute of Engineering & Technology, Patiala- 147004 (Punjab), India

I.INTRODUCTION

A.PURPOSE

To prepare an efficient algorithm to sort rice grains in with respect to the differences in grain size. The raw rice available in the market consists of anomalies in its size, shape etc. This algorithm generally focuses on the technique of sorting on the basis of the size. This system focused on the improvement in the quality of the rice available to the customers in the market. Although there exist some conventional methods for this task but automation is the need of the time.On removal of such imperfections, quality of rice will be improved which will further improve the food safety.

B. USE OF MATLAB SOFTWARE

It is a proprietary programming language developed by MathWorks. MATLAB allows a wide range of tasks as well as functions to be implemented. It incorporates matrix controls, plotting of functions and information, usage of algorithms, production of UIs and interfacing with projects written in different languages including C++, C, C#, Fortran, Java, and Python. Basically, theprovided image is converted into a matrix form on uploading it to software. Conversion to matrix form makes it easy to manipulate the data and operate it to provide efficient output.

C. QUALITY OF RAW RICE

Rice available in the market contains anomalies in its texture, size, shape, etc. Although at the time of harvesting the major anomalies are removed but the left aredifficult to handle. A lot of manpower is wasted in this task for which automation is needed. Especially in the Indian markets there are various types of rice available that includes, Regular long grain white rice, Brown whole grain, Basmati,Pusa, Laxmi Bhog, etc. Generally, these rice differ in terms of their grain sizes. That's why sorting on the basis of size will be extremely beneficial.

D. POTENTIAL FOR FOOD INDUSTRY

Food industry especially the Indian food industry is dependent on the labor up to a higher extent. This diminishes industry's efficiency. There is a high demand of automation in this industry. This

is the main reason of this paper to be focused on this type of industry. The sorting of rice provides a boost to this industry. Sorting removes the anomalies in the rice while improving the quality of rice. Improvement in rice quality would be beneficial for the consumers as they will get better quality for low prices.

II. FLOW CHART

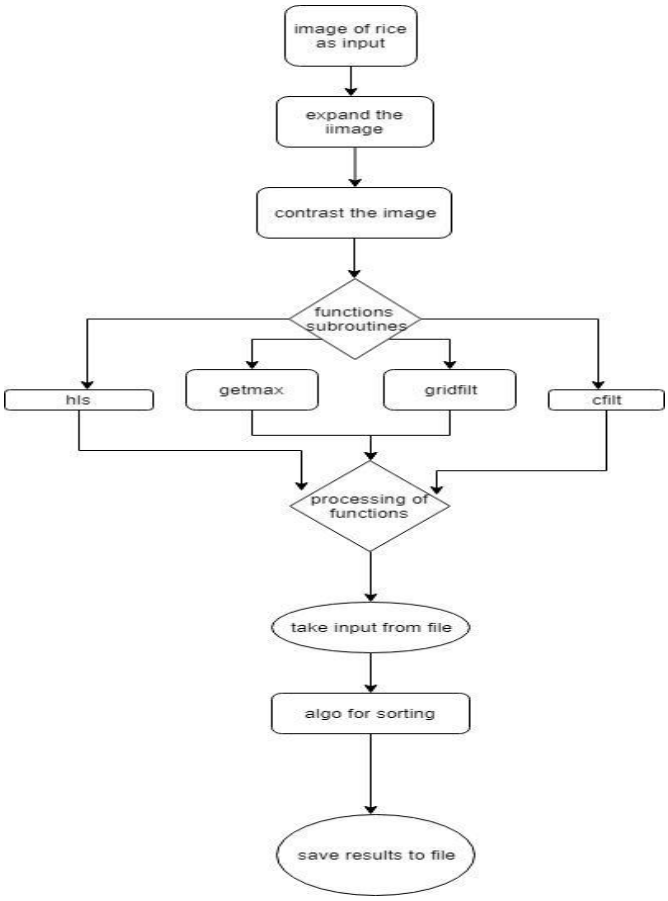

Figure 1: Flowchart

A.EXPLANATION OF THE FLOWCHART

The first is ‘expand’ which adds the black border around the image necessary for processing it with grid and circular filters. The second is ‘contrast’ which converts image into black and white image. The third, ‘getmax’, takes a gander at the image, and according to the assigned radius (r=5) counts the no. of pixels in (2r+1) size grid. Next, the ‘cfilt’ subroutine runs the image through the circular filter using the given radius size. 'Horizontal Layered Scanning (HLS)', as the name infers, runs the HLS counter once on the image after running through the circular filter, for an initial count.

The 'gridfilt' subroutine runs the image through, for this situation, 2 cycles of grid filtering. The algorithm at that point gets a new count on each cycle and, bearing the outcome doesn't differ

excessively a long way from the running average, includes it into thought for a last average. The average is then determined and, as should be obvious, there is an alternative to demonstrate the outcome in either correct or adjusted outcomes, for stylish reasons, contingent upon what result is wanted by just uncommenting the "total… " line.

The ‘gridfilt’ subroutine collects the real image and details like, image’s size, the desired grid size for the filter to utilize, and the circular filter radius. The radius is just considered with the goalthat the program doesn’t keep running on anything but the original image, barring the black border, which depends on the size of radius. A second matrix is made precisely equivalent to the given one, at that point the algorithm looks pixel by pixel at the original image. For those pixels, it takes a gander at the encompass zone utilizing the given grid, and tallies the quantity of white pixels. If the grid doesn’t meet the requirements (herelook for all white), then the pixel is turned black. Two matrices/images are used in order to make changes without disturbing the original image, which would affect the algorithm as it runs.

III. METHODS OF CLEANING

For cleaning of the image, the following filters are used: -

- Direct Contrast filter
- Grid filter
- Circular filter

IV. SPECIFICATIONS

A. Direct Contrast Filter

This filter converts the RGB image into Grayscale. MATLAB program can be used to convert the images in the grayscale format. Basically, there are two reasons for using this logic. First reason is that there is a three-dimensional matrix of RGB which decide the color of the image, performing gray scalinghelps to solve the problem in this paper. Second reason when there will be two colors in the end on which the rice count is to be performed. One important aspect in the whole process is that the camera should be in apposition so that there is no shadow on the image and lighting in the area where image is captured must be proper. And the other thing is that the camera must be held stable so that blurring doesn’t occur. Now talking about the threshold values for the image, if the value is 0 then pixel is said to be black and the if value is near to 255 then the pixel is black.

B. Grid Filter

The grid filter was created to take benefit from the black and white filtered images. In grid filter, the image is experienced pixel by pixel and find the matrix or grid that encompasses that pixel. However, if all pixels in the image are to be examined, a grid around the image's corner must be examined.

Then, the original image/matrix is reinserted inside a new image, focused within it. This outcomes in the real picture with a black outskirt drawn around the outside of it sufficiently expansive to run the algorithm over the whole real image.

When there is the bordered image, the grid filter algorithm runs, beginning at what was the first upper-left corner of the image, and goes pixel by pixel along the first row, then each succeeding row, until the bottom right corner of the image is reached. In the event that it discovers a white pixel, it takes a gander at the grid encompassing it, at that point, turns it dark if there are any dark pixels in that grid, otherwise keeping it white. This has the impact of leaving the "core" of the rice, while disposing of any overabundance information. While extremely successful, it could now and then be excessively ruinous, and can dividea rice into various rice, especially if the rice encapsulates any dark pixels.

Below is an example of the filter used on an early image:

Figure 2: Image showing coarse grain

Figure 3: Image showing fined grain

C. Circular Filter

It works under indistinguishable standard from the Grid Filter, yet takes a gander at a round region inside each grid, and takes into consideration a couple of dark pixels to be incorporated, keeping us from "splitting" the rice down the middle, as the grid filter some of the times did, just as better fitting the rice, then the square filter, because of the innate state of the rice. This averts unreasonable demolition generally, giving much better images. The circular filter, utilized without anyone else, gave the below result:

Figure 4: Image before circular filter

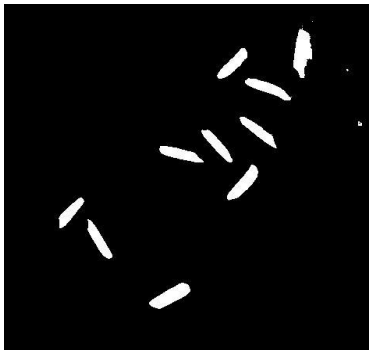

Figure 5: Image after circular filter

As should be obvious in these images, it leaves more the rice unblemished than did the Grid filter, in any case, you can likewise observe that a couple of spots of insignificant information need to stay in the image. Hence is the reason that the last filter includes the utilization of each of the three filters, first gray-scaling and utilizing the direct contrast filter, then utilizing the Circular filter to get the above image, at last utilizing the grid filter on a much lower level than it has been utilized beforehand, disposing of those last couple of bits of incorrect information that would harvest destruction in the counting algorithms, the last consequence of which can be seen below:

Figure 6: Output 1

Figure 7: Output 2

Figure 8: Output 3

As arduous as all of this cleaning and filtering was, it paled in comparison to the other half of the ultimate goal, the counting algorithms.

D. Horizontal Layered Scanning (HLS)

The essential thought is to take a gander at each row all in all, following what number of rice are gone through, and contrasting it with the previous row. When the algorithm notes that it is counting less rice than before, it makes the assumption that it has completed passing through a grain of rice. For example, let's look at the rice shown in the image below:

Figure 9: Input Image for HLS

Moving down, it can be seen that image there is only dark vacancy, so it peruses [0, 0], which means 0 rice in the row, 0 rice finished. When it reaches the row where the upper-right rice grain is placed, it peruses [1, 0], which means one rice in the line, and none finished, a similar when it reaches the second rice grain on the left [2, 0]. Before it reaches the last rice grain anyway, it initially recognizes the end of one rice grain, count [1,1]; one rice grain in the row, one completed.

A couple of rows later, it completes the second rice grain [2, 0]. Similarly, it counts the 3rd rice grain present in the image. This method is highly efficient and effective, particularly when compared to the overwhelming complexity of the border programs. As the issue lies with the positioning of rice, and position of rice grain need to be altered sometimes. Multiple iterations of the grid destructive filter are utilized so the rice which cause the issue are metaphorically "pulled apart".

Figure 10: Output 1 HLSFigure 11: Output 2 HLSFigure 12: Output 3 HLS

At that point, the average of the counts was taken which gave a progressively exact tally, expelling any numbers that exceedingly go amiss from the rest as a safety measure. This can occur for a situation where a image is excessively altogether wrecked, and rice grains is part into a few extensive spots, which would every at that point be considered its very own rice, misleadingly expanding the tally.

(IJIASE) 2018, Vol. No. 4, Jan-Dec **e-ISSN: 2454-9258, p-ISSN: 2454-809X**

V. AREA ESTIMATION FOR COUNTING PURPOSE

The essential thought behind this counting procedure is to estimate the quantity of rice present in a image by using the region that a solitary rice grain takes up in that image {noted by the variable (Rg)}.

$$Gr / Rg = Ms \qquad \text{---------- (A)}$$

Where, Gr= number of pixels in the image have a value comparing to a rice grain
Rg= region occupied by one rice grain
Ms= count of grains of rice in the image

Figure 13: Image for area estimation

There are numerous complexities involved amid interpretation from hypothesis into the Matrix Laboratory (MATLAB) construct. First, a method to calculate (Rg) must be devised; the number of pixels associated with a single rice grain. Few grains are identical so there is need to find the average pixel value for many rice grains. To do so, many images of individual rice grains were analyzed. All of the images were cleaned and gray scaled with a threshold to produce a binary image with values of 0 and 255; the latter is the pixel value associated with a grain and 0 is the pixel value of the background. This reduces the task to counting all pixels with the value of 255.

The result is a constant of Rg = 2,694 average pixels in one rice grain. The same program can be reused to count Gr for any image with any number of rice grains present. By using the formula (A) i.e. mentioned above, count of rice grains can be determined.

VI. ANALYSIS OF DATA

Taking an overall average of the data, 92.59 % accuracy was achieved. It wasn't until the very end that a filter is produced that, in tandem with the grid filter and contrast cleaning method, gave comparable results, independent of all third-party algorithms.

A few patterns additionally appear inside in the two arrangements of information. The most remarkable of these is the moderate, however relentless drop in precision as the quantity of rice increments. This is, obviously, because of the intrinsic imperfection of HLS as expressed

(IJIASE) 2018, Vol. No. 4, Jan-Dec **e-ISSN: 2454-9258, p-ISSN: 2454-809X**

beforehand. With more time, the database can be extended, including unmistakably more images per test size, and test thousands of images, rather than ceasing at the hundredth rice, as it would take a long time, given all rice grains are carefully checked for each and every image.

VII. SORTING ALGORITHM

Block /partitions of pixels in an image are made. It needs to be checked whether the rice is of the standard size or not, this is the main idea for performing this action in that loop. If the rice is above the threshold or just merely equal to the threshold then its output is termed as the fair rice and if the rice doesn't suffice the threshold size criterion then it is termed as a false/small rice which cannot be accommodated in the rest of full-length rice.

The code working in this loop is as follows: firstly, the per block pixel dimension of the image, this task was performed by the trial and error to approximate no. of pixels in a block of image. Now if there are some scattering of black particles in the image then this will be removed by the earlier mentioned filters. The erratic rice image sample is depicted using the 0 in the output. If per block size of the image is checked again then this result remains the same because the operation performed by us are of fixed threshold values. The time constraint in this problem is to perform the calculation in the certain time stipulation such that output must be computed by code in less than time as compared to the time of fall of the rice on the holes in the bottom of the machine. Hence the algorithm for computations of the rice size must be less than the time of fall, this condition is main condition before all condition has to be met in order to get the job done.

VIII. IEEE STANDARDS USED

IEEE standards that have been used in the paper are as follows:

- 610.4-1990 - IEEE Standard Glossary of Image Processing and Pattern Recognition Terminology.
- 208-1995 - IEEE Standard on Video Techniques: Measurement of Resolution of Camera Systems, 1993 Techniques.
- 1858-2016 - IEEE Standard for Camera Phone Image Quality

IX. RESULTS

Observations and Results

Sample Image and Output No.- 1

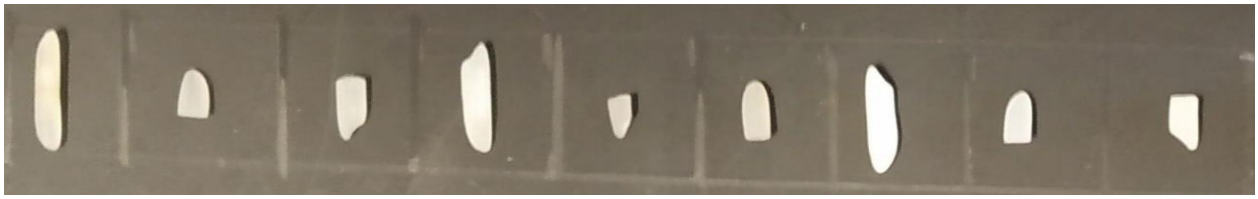

Figure 14: Input Image 1

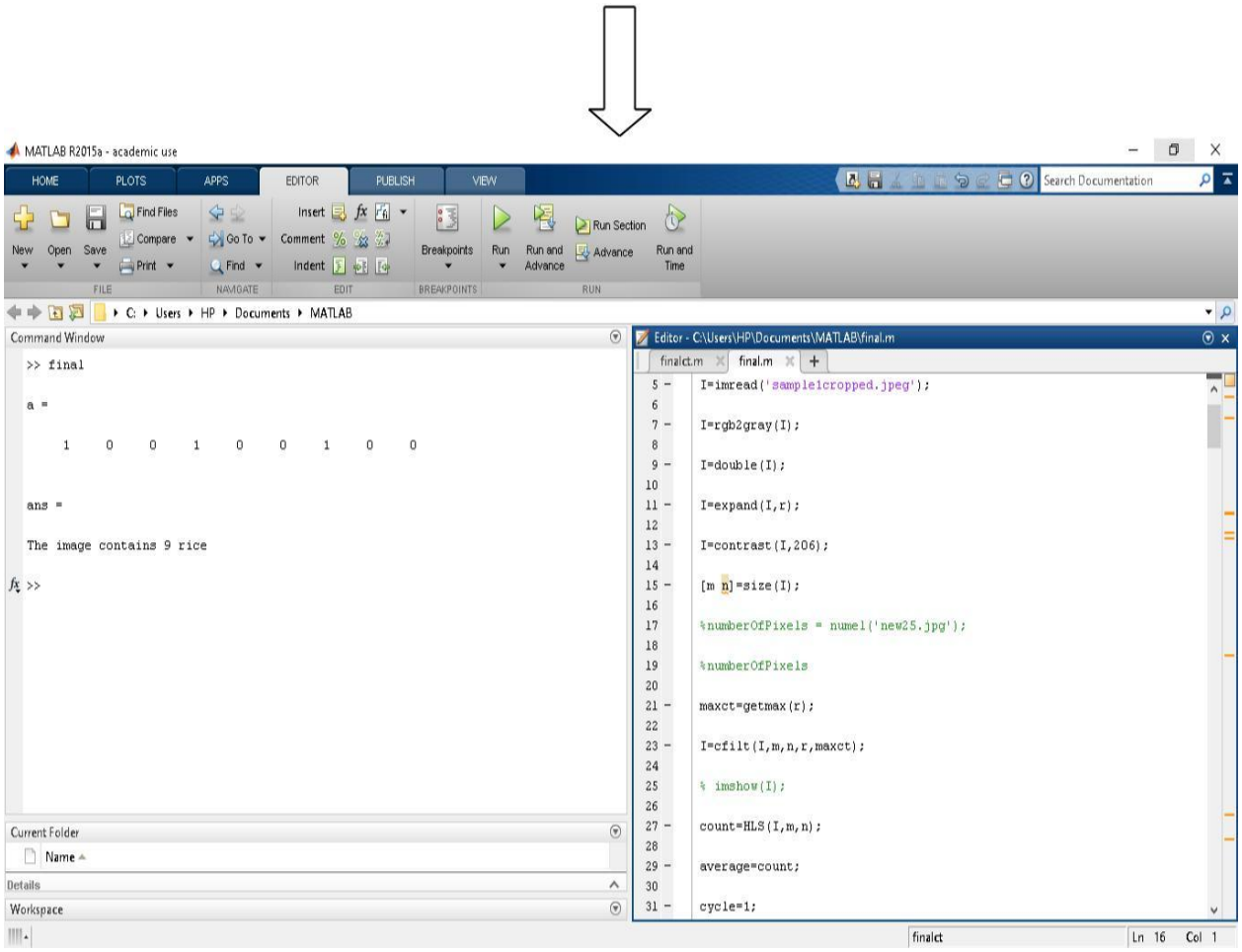

Figure 15: MATLAB Output 1

Sample Image and Output No.- 2

Figure 16: Input Image 2

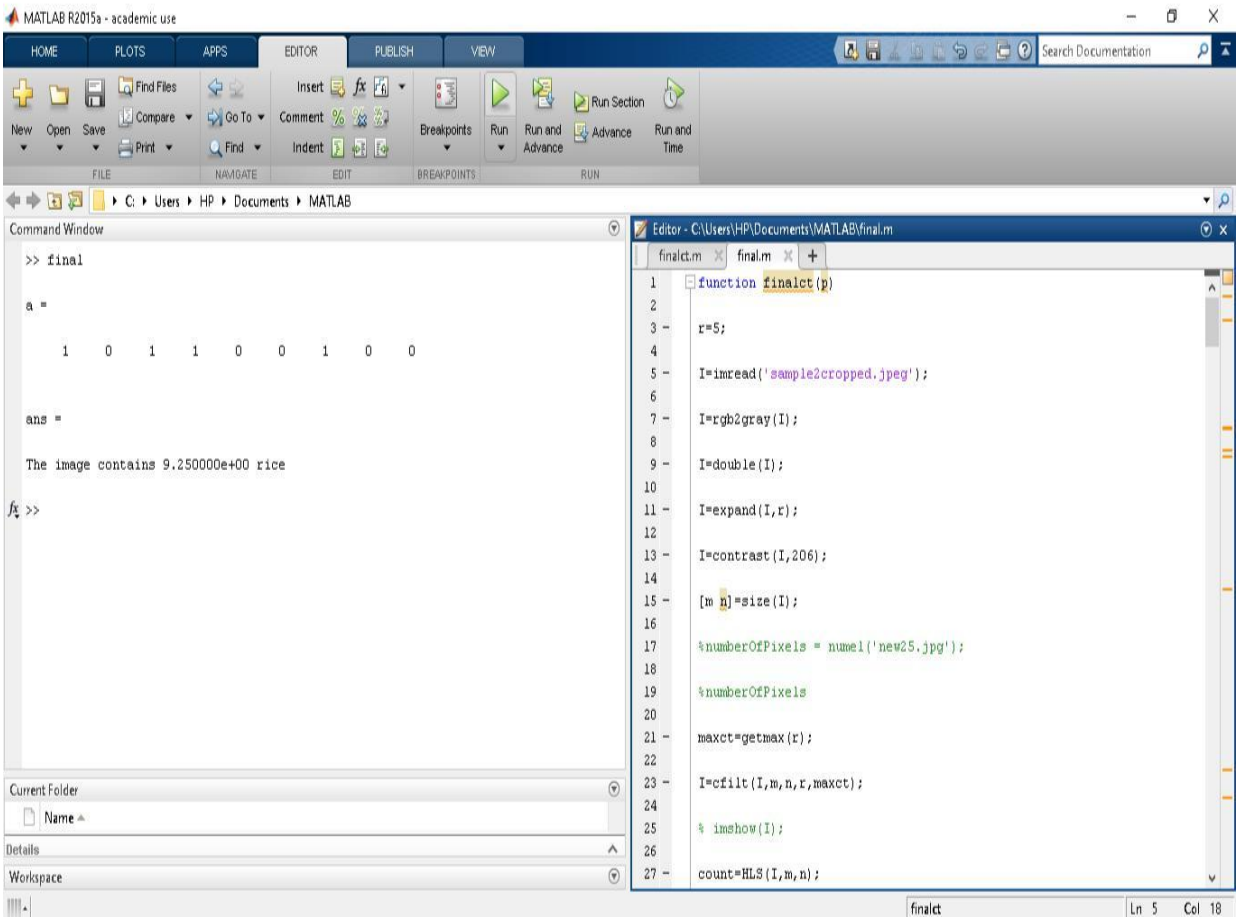

Figure 17: MATLAB Output 2

X. EFFICIENCY

$$\text{Efficiency}=\frac{Totalno.ofrice-No.ofrice\ havingerror}{Totalno.ofrice}\ x\ 100\ \%$$

Total Sample data of 15 images was observed. Each image was containing 9 rice grains. Then, by verifying each image carefully, it was found that 10 grains were inaccurately detected.

	A	B	C	D	E	F	G	H	I	J	K
1	1	0	0	1	0	0	1	0	0	sample1cropped.jpeg	
2	1	0	1	1	0	0	1	0	0	sample2cropped.jpeg	
3	0	1	1	1	1	1	1	1	1	sample3cropped.jpeg	
4	1	1	0	1	1	1	0	0	1	samplecropped10.jpeg	
5	1	1	1	1	0	0	1	0	1	samplecropped11.jpeg	
6	1	0	1	1	1	1	0	0	1	samplecropped12.jpeg	
7	1	1	1	1	0	1	1	1	0	samplecropped13.jpeg	
8	1	0	1	1	1	0	1	0	1	samplecropped14.jpeg	
9	1	1	1	0	0	1	1	1	0	samplecropped15.jpeg	
10	1	1	0	1	1	0	0	1	0	samplecropped4.jpeg	
11	0	0	0	0	1	0	1	0	1	samplecropped5.jpeg	
12	1	1	1	0	1	0	1	1	0	samplecropped6.jpeg	
13	1	0	1	1	1	0	1	1	1	samplecropped7.jpeg	
14	1	1	0	1	1	0	0	0	1	samplecropped8.jpeg	
15	1	1	0	1	1	0	0	0	1	samplecropped9.jpeg	

Figure 18: Results of 15 sample images stored in MS Excel

$$\text{Efficiency}=\frac{(15*9)-10}{(15*9)}\ x\ 100\ \%= 92.59\%$$