Available online www.jsaer.com

Journal of Scientific and Engineering Research, 2017, 4(1):1-12

Research Article

ISSN: 2394-2630
CODEN(USA): JSERBR

Development of A Mobile Application for Automated Paddy Grading

Mohammad Asaduzzaman Rasel[1*], Mohammad Osiur Rahman[2]

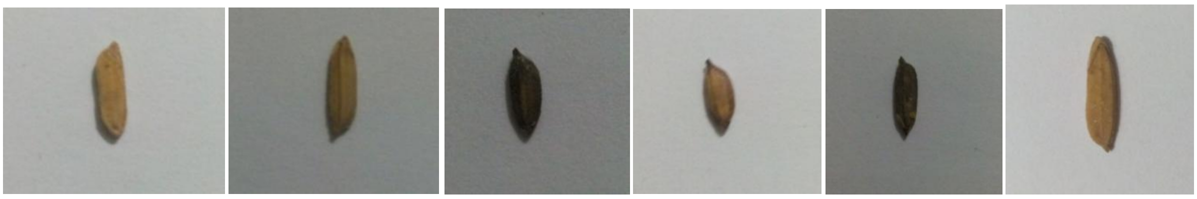
[1]Department of Information Technology, Malaysia University of Science and Technology, Kelana Square Block C, 17, Jalan SS 7/26, 47301 Petaling Jaya, Selangor, Malaysia
[2] Department of Computer Science and Engineering, University of Chittagong, Chittagong 4331, Chittagong, Bangladesh

Abstract Development of A Mobile Application for Automated Paddy Grading contains of modern Digital Image Processing Techniques for detecting various kinds of paddy grade. There are very few grading systems for paddy which are very lengthy process and expensive. But in this process, it is very easy to detect the grade of paddy and possible to get result immediately by using android device with low cost. For detecting grade of paddy it is important to capture the image of sample paddy from a certain distance to avoid error result. The size of picture is the main fact to grading paddy where a wrong size of picture can give an error result. Grade detection for different quality of paddy is really hard because of their different colors and sizes. By examining hundreds samples of paddy, it has been possible to get much the nearest accurate result. Always taking picture from a certain distance can minimize the error and can give an accurate result. Different kind of image processing operations are ran here for counting pixel of paddy's height, weight and for measuring the total size of paddy. By calculating height, weight and size, it is possible to detect the identity of paddy. Using the height-weight, size and color processing techniques, Total success Rates (TSR) are 99.5%, 99.6%, 99.7% and 99.8% respectively. The exceptional performance acquired from this grading system is the output of corroboration, which is the ideal among the restrictive methods of grading paddy.

Keywords Android apps; image processing; threshold; paddy color band; image acquisition

1. Introduction

There are huge works for paddy research system and detecting disease of paddy. Most of previous works are for lab performance. Very few works are for field server systems. And all works depend on digital image processing system. The digital image processing system can give us accurate result by filtering several steps. Consuming time is possible decrease the time duration between taking picture and image processing system. So that there has been chosen android based image processing system cause in android device, it is possible to get easily camera and processing system which help to get instant result.



(a) (b) (c) (d) (e) (f)

Figure 1: (a) Bini, (b) Bashmoti, (c) Kalo Zira, (d) Kalo Zira 2, (e) Kala Manik, and (f) B29

2. Paddy Grading

Rice seed is known as Paddy. Rice is the seed of the grass species Oryza sativa (Asian rice) or Oryza glaberrima (African rice). As a cereal grain, it is the most widely consumed staple food for a large part of the world's human population, especially in Asia. It is the agricultural commodity with the third-highest worldwide production, after sugarcane and maize, according to data of FAOSTAT 2012. Rice, a monocot, is normally grown as an annual plant, although in tropical areas it can survive as a perennial and can produce a ratoon crop for up to 30 years [1]. The rice plant can grow to 1–1.8 m (3.3–5.9 ft) tall, occasionally more depending on the variety and soil fertility. It has long, slender leaves 50–100 cm (20–39 in) long and 2–2.5 cm (0.79–0.98 in) broad. The small wind-pollinated flowers are produced in a branched arching to pendulous inflorescence 30–50 cm (12–20 in) long. The edible seed is a grain (caryopsis) 5–12 mm (0.20–0.47 in) long and 2–3 mm (0.079–0.118 in) thick [3]. There are more than 15000 kinds of rice and more than 15000 rice seeds which are paddy in the whole world [2]. Some paddy’s images are shown in figure 1.

3. Development of the Proposed System

Automated Paddy Grading System is basically consists of two parts that are measuring height-weight, and counting total size of paddy. It is the common to take picture in a fixed size for the two parts. The proposed diagram of paddy grading system is shown in figure 2.

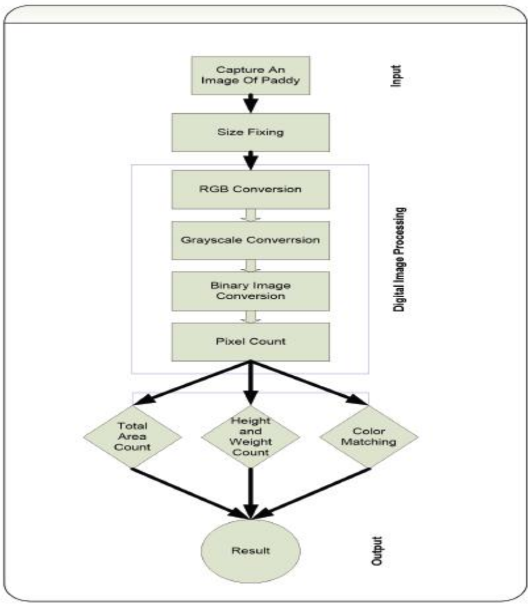


Figure 02: The Proposed Diagram of Paddy Grading System

The proposed input and output process of the mobile application for automated paddy grading is given in figure 3. Different kinds of paddy image will be the input and by completing the proposed operation final output will be achieved.

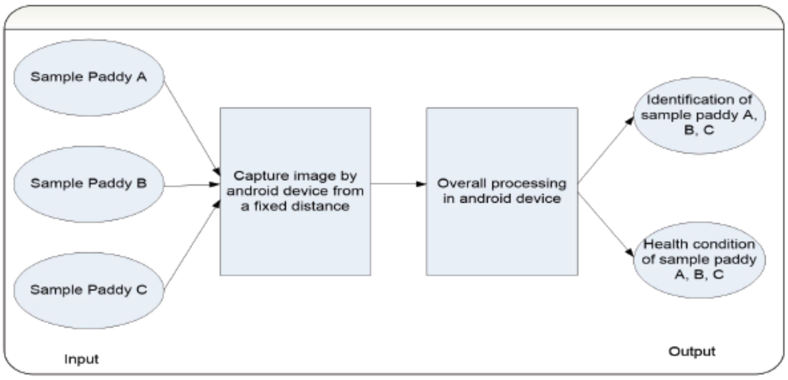

Sample Paddy A
Sample Paddy B
Sample Paddy C
Capture image by android device from a fixed distance
Overall processing in android device
Identification of sample paddy A, B, C
Health condition of sample paddy A, B, C
Input
Output

Figure 3: The Proposed Diagram of Input and Output Process of Paddy Grading System

3.1. Algorithm

At first there will be binary conversion from bitmap image to binary image to get every pixel position by threshold ranging [4]. The proposed binary conversion is given below as like as pseudocode implementation.

```
Bitmap bmpBinary = Bitmap.createBitmap(bmpOriginal);
for( x = 0; x < width; ++x)
    for( y = 0; y < height; ++y)
        get one pixel color
        pixel = bmpOriginal.getPixel(x, y);
        red = Color.red(pixel);
        get binary value
        if(red < threshold)
            bmpBinary.setPixel(x, y, 0xFF000000);
        else bmpBinary.setPixel(x, y, 0xFFFFFFFF);
```

After finishing binary conversion from bitmap image, there another operation will run for pixel count and pixel position calculation for identifying paddy species and to get final output.

```
Bitmap bmpvalue = Bitmap.createBitmap(bmp);
for( x = 0; x < width; ++x)
    for( y = 0; y < height; ++y)
        pixel = bmp.getPixel(x, y);
        red = Color.red(pixel);
        if(red == 0)
            if(start==true)
                start = false;
                startX = x;
                startY = y;
            endX = x;
            endY = y;
x1 = startX - endX;
y1 = startY - endY;
dx = Math.pow(x1, 2);
dy = Math.pow(y1, 2);
distance = Math.sqrt(dx+dy);
```

Finally, after finishing all those steps, output will tell the identity of paddy with their height-width and total pixel number as total paddy size.

3.2. Image Acquisition Technique

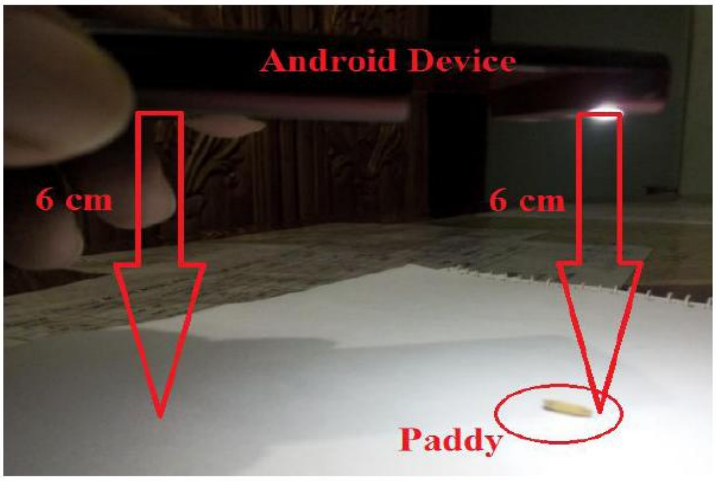


Figure 4: Image acquisition technique

To take a picture of sample paddy, there has been used three brands mobile phone. They are Samsung S DUOS GT-7562, Walton Primo HM and Sony. There has to maintain a constant distance which is 6 cm from sample paddy to mobile camera. To maintain this distance user can use an object which has 6 cm height and have to keep the mobile phone over that object so that the distance value will be kept in constant. Image acquisition technique is shown in figure 4.

Cause during the capture, human hand can remove little from target distance or it can be jerked. And always have to be careful about the light intensity; it cannot to be too light or dark. User has to also careful about the shadow of hand. During the taking picture, have to ensure about no shadow and has to use white background like paper or white board. For understanding, there is an image of taking picture. There has maintained 6 cm distance and proper light intensity for perfect result.

3.3. Image Size

There is a number of ways to do this. The most obvious, because here is using the image preview function of the camera, is to set an appropriate preview size [5]. Change the *switchcamera ()*method of *DetectPaddyActivity* to the version used in *ManageCameraFastActivity*:

```
        Camera.Parameters cameraParameters = mCamera.getParameters();
        List<Size> sizes  =  cameraParameters.getSupportedPreviewSizes();
        int width = Integer.MAX_VALUE, height = Integer.MAX_VALUE;
        for (int i=0; i<sizes.size();i++){
                Size s = sizes.get(i);
                if (s.width <width) {
                   width = s.width;
                   height = s.height;
        }
}
cameraParameters.setPreviewSize(width, height);
mCamera.settParameters(cameraParameters);
```

3.4. Bitmap to Binary Conversion

According to X and Y direction, every pixel will be detect and match with RGB color value. It will be start from (0, 0) position of (x, y). The loop will be continue until the last position of pixel in selected image. Suppose, image size is 420 x 840, then last position will be (420, 840) of (x, y). Have to limit the range of x, y within bitmap. Then, if x < 0; x will be 0; Otherwise x > bitmap.getWidth()-1; then x = bitmap.getWidth()-1. On the other hand, if y < 0; y will be 0; Otherwise y > bitmap.getWidth()-1; then y = bitmap.getWidth()-1. After converting in bitmap, touching position's pixel will be match with RGB color value [6]. In figure5, (a) and (c) are Bitmap images are shown while in the same figure, (b) and (d) are Binary images of (a) and (c) are shown.

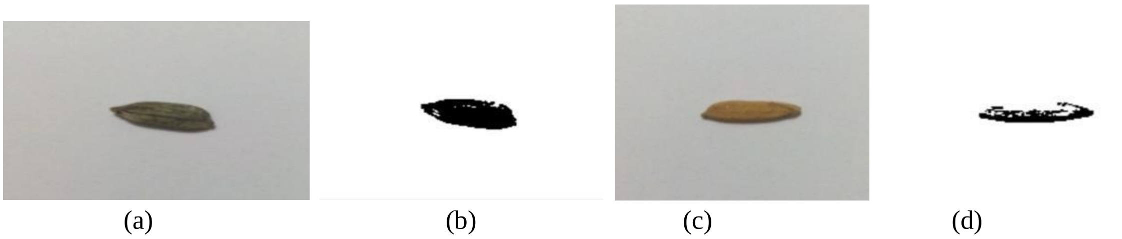
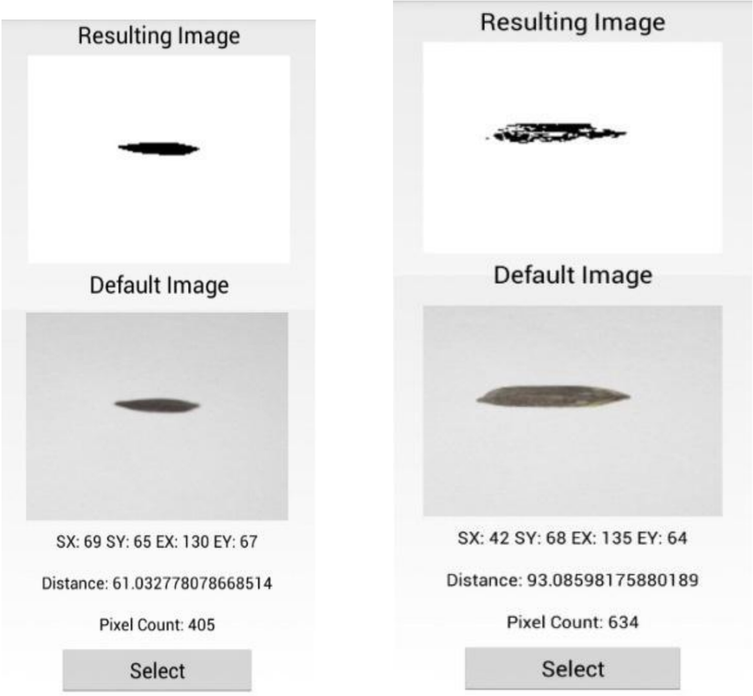


Figure 5: (a) and (c) are Bitmap Images, (b) and (d) are Binary Images of (a) and (c)

After completing conversion of binary image, capture image will be the combination of 0 and 1 [7]. In the picture, paddy sample will be replaced a certain number of pixel. Those pixel will be counted by x and y direction in image. Pixel counting loop will be ran from (0, 0) position of (x, y). Where 1 will be found, that pixel will be gone in stack. Loop will be continued until the last pixel of 1in the replaced area of paddy. Then after finishing the loop, from the stack total number of pixel will be counted. By this way, total pixel of area we can get it which is paddy's. For different kind of paddy like Bashmoti, Kalozira, Kalozira 2, B29, Bini etc. the total pixel of area will be different. So it has to convert main image to Grayscale, then have to convert to Binary image, then get value and finally have to set Bitmap image. Then according to x and y, loop will search 1 and then that pixel will send in stack. Finally it is possible to get total pixel. The whole processing will run like below statement [8, 9]:

```
Bitmap image1 = (Bitmap) data.getExtras().get("data");
defaultImage.setImageBitmap(image1);
image1 = toGrayscale(image1);
image1 = toBinary(image1);
image1 = toValue(image1);
captureImage.setImageBitmap(image1);
```

In figure 06, the total pixel count of (a) Kalamanik Paddy and (b) Kalozira Paddy are shown.



SX: 69 SY: 65 EX: 130 EY: 67

Distance: 61.032778078668514

Pixel Count: 405

SX: 42 SY: 68 EX: 135 EY: 64

Distance: 93.08598175880189

Pixel Count: 634

(a) (b)

Figure 06: Total pixel count of (a) Kalamanik Paddy and (b) Kalozira Paddy

3.5. Pixel Count

For measuring paddy height-width and total pixel number, pixel counting is necessary. So that in figure 7, (a) has shown the pixel positions according to height – width of a bitmap image. Every square box is consider as a pixel and for counting pixel have to count each square box. And also in figure 07, (b) has shown the pixel position in X and Y axis as like as graphical view. From this graph, it is easy to measure pixel position and pixel number according to X and Y axis[10]. So, it can be written that:

1. startX = x;
2. startY = y;
3. endX = x;
4. endY = y;
5. x1 = startX - endX;
6. y1 = startY - endY;
7. dx = Math.pow(x1, 2);
8. dy = Math.pow(y1, 2);
9. distance = Math.sqrt(dx+dy);

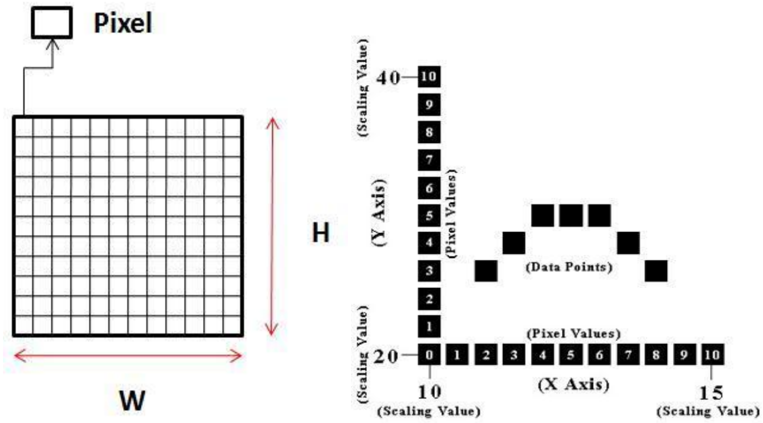


Figure 7: (a) According to height – width pixel positions in a Bitmap Image; (b) According to X – Y axis pixel positions

4. Experimental Result

4.1. Reference Database

There has created a database according to paddy height-width and total pixel number.

Table 1: Reference database of paddy grade

Category No.	Grading	Height	Width	Pixel Number
1	Kalo Zira	52 – 53	26 – 27	712 – 720
2	Kalo Zira 2	52 – 53	27 – 28	622 – 630
3	Kala Manik	67 – 69	33 – 36	1100 – 1200
4	Bashmoti	50 – 53	27 – 29	313 – 320
5	B29	75 – 80	34 – 37	541 – 550
6	Bini	53 – 55	22 – 23	465 – 470

For measuring, there are some paddy samples data in table 02.

Table 02: Paddy Sample Measurement

Sample No.	Height	Width	Pixel Number
1	52	27	713
2	53	28	625
3	68	34	1150
4	52	28	317
5	77	36	545
6	53	23	469

4.2. Matching Score and Making Decision

Now we are going to compare with table 1 and table 2. In table 1, there is the database of reference paddy grade and in table 2; there is the database of sample paddy's height-width and total pixel number. Now, by comparing these two tables it is possible to get the matching score and easy to make a decision. Sometimes, according to database one sample of paddy can be matched with 2 or 3 kinds of paddy grade. In that case, it is decided by using kNN method [11, 12].

Table 3: Matching score and making decision

Sample No.	Matching height with category	Matching width with category	Matching pixel number with category	Overall matching with category
1	1,2	1,2	1	1
2	1,2	2	2	2
3	3	3,5	3	3
4	1,2,4	2,4	4	4
5	5	3,5	5	5
6	1,2,4,6	6	6	6

Table 4: 100 samples paddy compare with reference database and comparison between estimated height and referenced height.

Sample No.	Name	Referenced Height	Width	Total Pixel	Estimated Height	Relative Error (%)	Paddy Color
001	B29	75	34	545	75	0.00	Golden
002	B29	78	35	543	79	2.08	Golden
003	B29	77	34	549	78	2.13	Green
004	B29	79	36	550	80	2.04	Golden
005	B29	80	37	550	80	0.00	Golden
006	B29	79	34	549	80	2.13	Green
007	B29	77	35	542	77	0.00	Green
008	B29	76	36	544	75	1.92	Golden
009	B29	79	37	546	80	2.00	Golden
010	B29	80	37	542	80	0.00	Golden
011	B29	78	36	541	79	2.08	Golden
012	B29	79	35	547	80	2.04	Golden
013	B29	77	34	548	78	2.13	Green
014	B29	77	35	549	78	2.13	Green
015	B29	80	36	549	80	0.00	Golden
016	B29	79	37	550	79	0.00	Golden
017	B29	79	36	549	79	0.00	Golden
018	B29	78	37	549	78	0.00	Green
019	B29	79	35	550	79	0.00	Golden
020	B29	80	34	550	80	0.00	Golden
021	Kala Manik	67	36	1137	67	0.00	Black Golden
022	Kala Manik	67	33	1200	68	3.33	Black Golden
023	Kala Manik	67	35	1126	67	0.00	Black Golden
024	Kala Manik	68	33	1112	68	0.00	Black Golden
025	Kala Manik	67	34	1129	67	0.00	Black Golden
026	Kala Manik	68	34	1128	67	3.22	Green
027	Kala Manik	69	33	1138	69	0.00	Black Golden
028	Kala Manik	68	34	1139	68	0.00	Black Golden
029	Kala Manik	69	33	1136	69	0.00	Black Golden
030	Kala Manik	67	34	1127	68	2.86	Black Golden
031	Kala Manik	67	36	1186	67	0.00	Black Golden
032	Kala Manik	67	33	1140	68	2.94	Black Golden
033	Kala Manik	69	35	1196	69	0.00	Green
034	Kala Manik	68	33	1128	69	2.77	Black Golden
035	Kala Manik	67	34	1139	67	0.00	Black Golden
036	Kala Manik	67	34	1149	68	2.94	Black Golden

037	Kala Manik	69	33	1176	69	0.00	Black Golden
038	Kala Manik	67	36	1166	67	0.00	Green
039	Kala Manik	67	33	1178	68	2.77	Black Golden
040	Kala Manik	68	34	1189	69	3.03	Black Golden
041	Kalo Zira	53	26	713	53	0.00	Black
042	Kalo Zira	52	27	714	52	0.00	Black
043	Kalo Zira	52	26	715	52	0.00	Green
044	Kalo Zira	52	26	716	53	2.04	Black
045	Kalo Zira	52	27	718	53	1.88	Black
046	Kalo Zira	52	27	720	53	2.22	Black
047	Kalo Zira	52	27	715	53	2.12	Black
048	Kalo Zira	52	26	717	53	1.85	Black
049	Kalo Zira	52	26	716	53	2.17	Black
050	Kalo Zira	52	27	715	52	0.00	Black
051	Kalo Zira	52	27	717	53	2.38	Green
052	Kalo Zira	53	26	713	52	1.88	Black
053	Kalo Zira	52	26	719	52	0.00	Black
054	Kalo Zira	52	27	718	53	2.33	Green
055	Kalo Zira	52	26	720	52	0.00	Green
056	Kalo Zira	52	26	715	53	2.08	Black
057	Kalo Zira	52	27	716	53	2.27	Green
058	Kalo Zira	53	27	719	53	0.00	Black
059	Kalo Zira	52	26	720	52	0.00	Black
060	Kalo Zira	52	27	714	53	2.17	Black
061	Kalo Zira 2	52	27	629	52	0.00	Green
062	Kalo Zira 2	53	28	630	52	2.85	Black
063	Kalo Zira 2	53	28	630	52	3.12	Black
064	Kalo Zira 2	53	27	630	52	2.77	Black
065	Kalo Zira 2	52	28	630	52	0.00	Black

066	Kalo Zira 2	52	27	629	53	3.33	Green
067	Kalo Zira 2	52	28	622	52	0.00	Black
068	Kalo Zira 2	53	27	622	53	0.00	Black
069	Kalo Zira 2	53	28	627	53	0.00	Black
070	Kalo Zira 2	52	28	627	52	0.00	Green
071	Kalo Zira 2	52	27	629	53	2.70	Black
072	Kalo Zira 2	53	27	628	53	0.00	Black
073	Kalo Zira 2	53	28	624	53	0.00	Green
074	Kalo Zira 2	52	27	624	53	2.77	Black
075	Kalo Zira 2	52	28	622	53	3.22	Green
076	Kalo Zira 2	53	27	630	52	0.00	Green
077	Kalo Zira 2	53	28	627	53	0.00	Black
078	Kalo Zira 2	52	27	627	53	3.22	Black
079	Kalo Zira 2	53	27	622	53	2.77	Black
080	Kalo Zira 2	52	28	622	53	3.23	Black
081	Bini	54	22	466	54	0.00	Golden
082	Bini	53	23	465	53	0.00	Golden
083	Bini	53	22	468	54	2.70	Green
084	Bini	55	23	469	55	0.00	Golden
085	Bini	54	22	470	54	0.00	Golden
086	Bini	53	22	469	54	2.44	Golden
087	Bini	53	22	467	55	2.63	Green
088	Bini	53	23	468	55	2.70	Green
089	Bini	54	23	466	54	0.00	Golden
090	Bini	55	22	465	54	2.56	Green
091	Bini	55	22	470	55	0.00	Golden
092	Bini	54	23	469	55	2.50	Golden
093	Bini	53	23	468	54	2.70	Green
094	Bini	53	23	467	53	0.00	Golden
095	Bini	54	22	466	54	0.00	Golden
096	Bini	55	22	465	55	0.00	Golden
097	Bini	54	23	467	54	0.00	Green
098	Bini	53	23	470	53	0.00	Golden
099	Bini	53	22	470	54	2.70	Green
100	Bini	55	22	465	55	0.00	Green

For measuring FRR and FAR, have to follow the equations which are shown in below.

$$FRR = \frac{Number\ of\ rejected\ genuine\ claims}{Total\ number\ of\ genuine\ access} * 100\% \qquad (i)$$

$$FAR = \frac{Number\ of\ accepted\ imposter\ claims}{Total\ number\ of\ imposter\ access} * 100\% \qquad (ii)$$

Table 05: Result of final testing of performance of the system for genuine

Threshold	Sample	Success	Reject	FRR %	Success Rate %
0	10000	200	9800	98	2
5	10000	500	9500	95	5
10	10000	1000	9000	90	10
15	10000	1000	9000	90	10
20	10000	2000	8000	80	20
25	10000	2000	8000	80	20
30	10000	3000	7000	70	30
35	10000	3000	7000	70	30
40	10000	4000	6000	60	40
45	10000	4000	6000	60	40
50	10000	5000	5000	50	50
55	10000	5000	5000	50	50
60	10000	6000	4000	40	60
65	10000	6000	4000	40	60
70	10000	7000	3000	30	70
75	10000	7000	3000	30	70
80	10000	8200	1800	18	82
85	10000	8200	1800	18	82
90	10000	9300	700	7	93
95	10000	9300	700	7	93
100	10000	9600	400	4	96

Table 06: Result of final testing of performance of the system for imposter

Threshold	Sample	Success	Reject	FRR %	Success Rate %
0	10000	9800	200	2	98
5	10000	9500	500	5	95
10	10000	9000	1000	10	90
15	10000	9000	1000	10	90
20	10000	8000	2000	20	80
25	10000	8000	2000	20	80
30	10000	7000	3000	30	70
35	10000	7000	3000	30	70
40	10000	6000	4000	40	60
45	10000	6000	4000	40	60
50	10000	5000	5000	50	50
55	10000	5000	5000	50	50
60	10000	4000	6000	60	40
65	10000	4000	6000	60	40
70	10000	3000	7000	70	30
75	10000	3000	7000	70	30
80	10000	2000	8000	80	20
85	10000	2000	8000	80	20
90	10000	1000	9000	90	10
95	10000	1000	9000	90	10
100	10000	800	9200	92	8

Due to several reasons, this assumption isn't true for real world paddy disease recognition systems. In some cases impostor patterns generate heights and weights that are higher than the heights and weights of some paddy patterns. For that reason it is a fact, that however the classification threshold is chosen, some classification errors occur. False reject and false accept resulting graph is shown in figure 8.

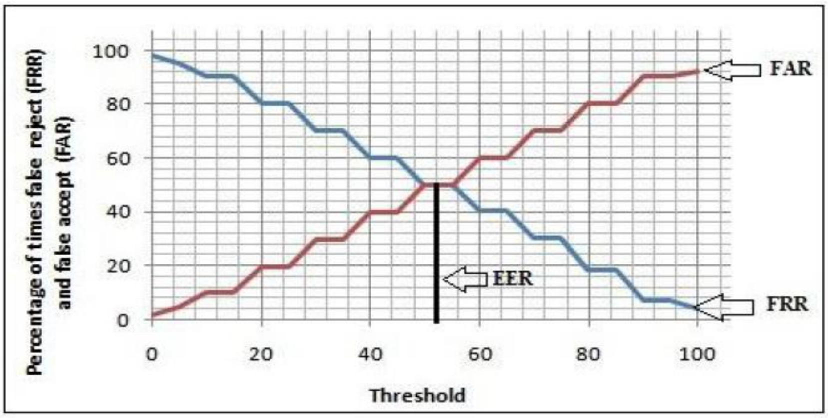


Figure 8: False reject and False accept resulting graph

Finally, according to percentage of FAR, we have to find out the TSR. So that in the below, TSR equation is given.

$$TSR = (1 - \frac{FAR + FRR}{Total\ number\ of\ access})*100\% \qquad (iii)$$

Table 7: Result of final testing of performance of the system for TSR (Hit Ratio)

FAR %	TSR %
0	0
2	90.0
5	95.5
10	96.5
20	97.5
30	98.5
40	99.5
50	99.5
60	99.5
70	99.5
80	99.6
90	99.7
92	99.8

From table 7, there are FAR and TSR value in data table. So, it is easy to draw the Receiver Operating Characteristic (ROC) curve. It is also known as TSR (Hit Ratio) vs FAR graph which is shown in figure 9.

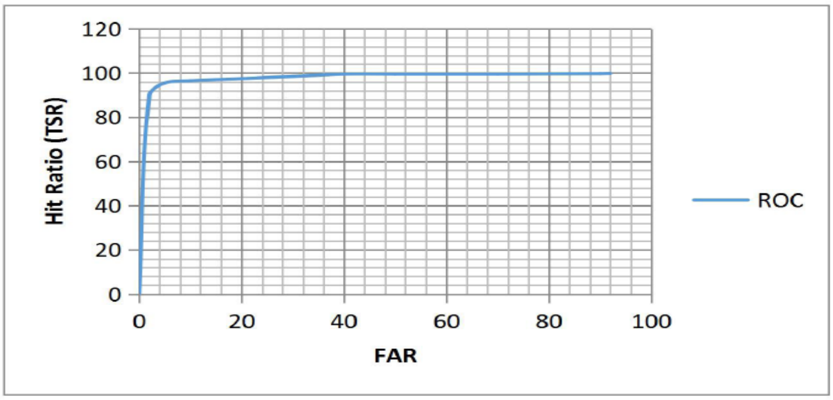


Figure 9: TSR (Hit Ratio) vs. FAR graph

5. Conclusion

The Automated Paddy Grading System is able to detect the identity of paddy and after getting result of identification, by color detecting and matching it can provide the result of paddy grading. For different kinds of paddy it can work successfully and it is able to help the farmers and the agriculture officer in a short time where the present technology takes huge time and cost. At last it can say that the whole process of the Mobile Application for Automated Paddy Grading has worked properly and the relative error is too small for the initial stage. More contribution and development of the process can give it more accuracy in detecting paddy grade and disease. However, the whole process of recognition has successfully established the advantages of modern technology in mobile application. The most important thing to say, this application can give us immediate result when we need to know the grade of paddy. On the other hand, it is able to work for not only paddy but also different kinds of crops by a small modification. In a scene, these techniques of detection can give the flavor of modern technologies to the people of whole world.

Acknowledgment

We are really grateful to Bangladesh Rice Research Institute (BRRI) and Botany department of University of Chittagong for giving us information which was necessary for our research work.

References

[1]. H.C.D. Wijayawardhana, H.M.V.G. Herath, P.A. Weerasinghe, & H.M.D.A.K. Herath. (2015). Morphological Variation in Selected Sri Lankan Rice (Oryza sativa L.) Accessions in Relation to the Vegetative Parameters. *Tropical Agricultural Research*, 26 (2): 380-389.

[2]. S. Hussain, T. Fujii, S. McGoey, M. Yamada, M. Ramzan, & M. Akmal. (2014). Evaluation of Different Rice Varieties For Growth And Yield Characteristics. *The Journal of Animal & Plant Sciences*, 24(5):1504-1510.

[3]. L. Vithyashini, & H.A.M. Wickramasinghe.(2015). Genetic Diversity of Seed Storage Proteins of Rice (Oryza sativa L.) Varieties in Sri Lanka. *Tropical Agricultural Research*, 27 (1): 49-58.

[4]. T. Meyyappan, SM.Thamarai, & N.M.Jeya Nachiaban. (2011). Lossless Digital Image Compression Method For BITMAP Images. *The International Journal of Multimedia & Its Applications (IJMA)*, 3(4):83-92.

[5]. Jaromir Przybylo, Marcin Piatck, Mariusz Pauluk, & Jakub Galicki. (2011). Low-Cost Mobile Image Processing Platform. *AUTOMATYKA*, 15(3):235-242.

[6]. T. Reps, S. Horwitz, & M. Sagiv. (1995). Precise interprocedural dataflow analysis via graph reachability. *In POPL '95: Principles of programming languages*, 22(1):49-61.

[7]. Richard F. Lyon. (2006). A Brief History of 'Pixel'. *Digital Photography II: IS&T/SPIE Symposium on Electronic Imaging, San Jose, California, USA*, 23(2):1-16.

[8]. Greg Milette, & Adam Stroud. (2013). *Professional android sensor programming*. John Wiley & Sons, Inc., 4th Ed., 280-320.

[9]. David G. Lowe. (2004). Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, 60(2):91-110.

[10]. Mausel, P.W., Kramber, W.J., & Lee J.K. (1990). Optimal band selection for supervised classification of multispectral data. *J. Photogramm. Eng. Remote Sens.*,56(1):55-60.

[11]. Woebbecke D.M., Meyer G.E., Von Bargen K., & Mortensen D.A. (2001). Shape features for identifying young weeds using image analysis. *Transactions of the ASAE 1995*, 38(1):271-281.

[12]. Meyer G.E., Mehta T., Kocher M.F., Mortensen D.A., & Samal A. (1998). Textural imaging and discriminant analysis for distinguishing weeds for spot spraying. *Transactions of the ASAE 1998*, 41(4):1189-1197.