Segmentation of Mushroom and Cap Width Measurement Using Modified K-Means Clustering Algorithm

Eser SERT , Ibrahim Taner OKUMUS

Computer Engineering Department, Engineering and Architecture Faculty, Kahramanmaras Sutcu Imam University, Avsar Kampusu 46100 Kahramanmaras, Turkey

esersert@ksu.edu.tr, iokumus@ksu.edu.tr

Abstract. *Mushroom is one of the commonly consumed foods. Image processing is one of the effective way for examination of visual features and detecting the size of a mushroom. We developed software for segmentation of a mushroom in a picture and also to measure the cap width of the mushroom. K-Means clustering method is used for the process. K-Means is one of the most successful clustering methods. In our study we customized the algorithm to get the best result and tested the algorithm. In the system, at first mushroom picture is filtered, histograms are balanced and after that segmentation is performed. Results provided that customized algorithm performed better segmentation than classical K-Means algorithm. Tests performed on the designed software showed that segmentation on complex background pictures is performed with high accuracy, and 20 mushrooms caps are measured with 2.281 % average relative error.*

Keywords

K-Means clustering, mushroom cap measurement, mushroom image segmentation.

1. Introduction

Due to their high nutritive content, mushrooms are one of the commonly consumed foods. Image processing techniques can be used in classifying, quality control and determining the size of a mushroom.

In the system, at first image is preprocessed and after preprocessing segmentation is performed with k-means clustering algorithm. In image processing, it is important to correctly selecting these steps, and successful application is very important to achieve the appropriate result.

In Fig. 1, a comparison of classical color k-means algorithm and proposed color k-means algorithm is provided. Algorithms are tested on 3 different mushroom types and as can be seen from the figure, proposed algorithm resulted in more successful segmentation. Figure also shows that unpreprocessed image segmentation with classical color k-means algorithm causes some problems.

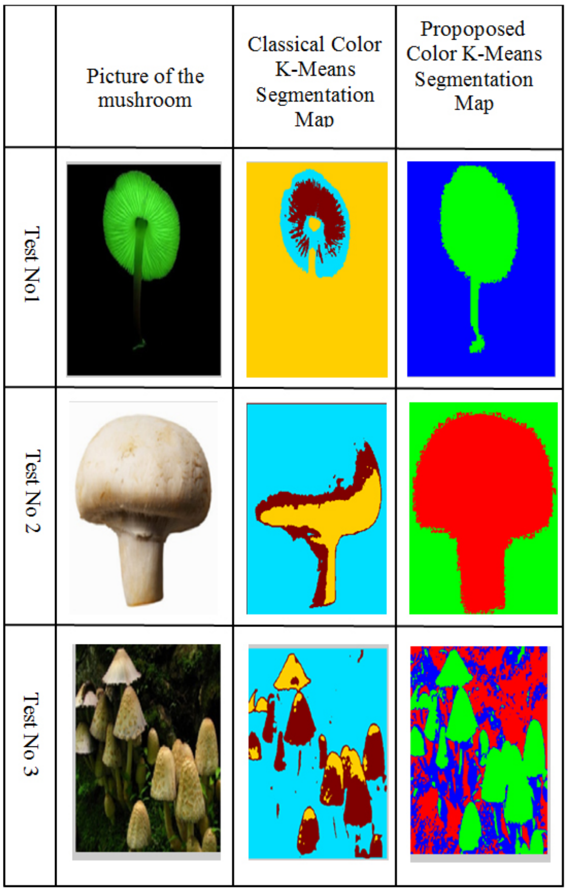

Picture of the mushroom
Classical Color K-Means Segmentation Map
Propoposed Color K-Means Segmentation Map
Test No1
Test No 2
Test No 3

Fig. 1: Comparison of classical color K-Means and proposed color K-Means algorithms.

For the segmentation process methods such as neural networks, support vector machine, genetic algorithms can also be used. However, segmentation can be performed using k-means clustering in a simple and effective way. In the literature Hong Yao [1] successfully applied the segmentation on a fish picture. Yong Zhang [2] applied the segmentation using PSO and PCM. Zhiqiang Lao [3] segmented White matter lesions using support vector machine.

2. Working Principle of the System

System contains 3 steps as can be seen in Fig. 2. First step is preprocessing, second step is segmentation and last step is showing results.

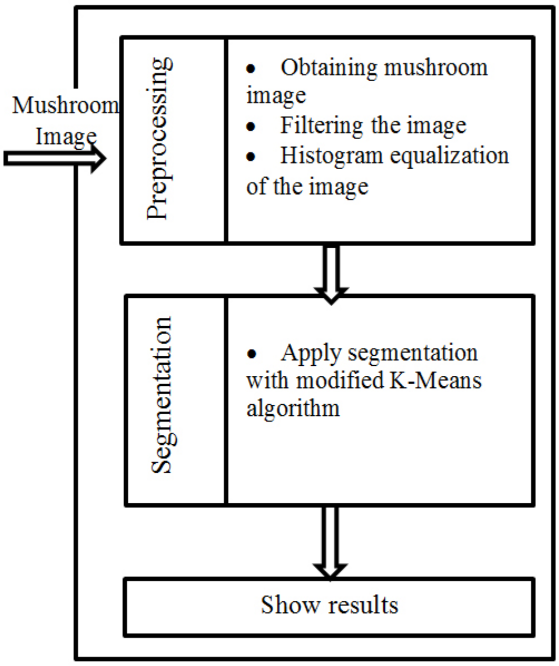

Mushroom Image
Preprocessing
• Obtaining mushroom image
• Filtering the image
• Histogram equalization of the image
Segmentation
• Apply segmentation with modified K-Means algorithm
Show results

Fig. 2: System diagram.

2.1. Preprocessing Step

In the beginning of this step mushroom picture is obtained, then filtering and histogram equalization is applied to the image.

1) Filtering Process

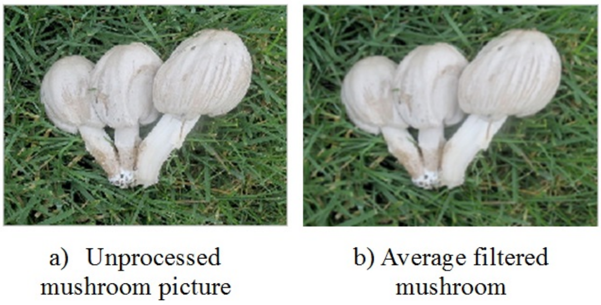
In the beginning, the obtained mushroom picture is turned into greyscale format. After that 3 × 3 average filter is applied to the picture. The reason for applying average filter is providing a smooth transition among the pixels to prevent noise. Eq. (1) shows the application equation of average filter [4]:

$$f(x,y) = \frac{1}{m \times n} \sum_{(s,t)\in Sxy} \text{gray\_pixel}(s,t). \quad (1)$$

In the equation, m and n adjust the width of the region that the filtering will be applied. gray_pixel (s,t) is the intensity level at point (s,t). Since 3 × 3 filtering is used in the process, Eq. (2) shows the same equation for 3 × 3 filtering:

$$f(x,y) = \frac{1}{3 \times 3} \sum_{(s,t)\in Sxy} \text{gray\_pixel}(s,t). \quad (2)$$

Figure 3(a) show unprocessed mushroom picture and Fig. 3(b) shows filtered mushroom picture.


a) Unprocessed mushroom picture

b) Average filtered mushroom

Fig. 3: Applying average filtering to mushroom image.

2) Histogram Equalization

After filtering, histogram equalization is applied to the image. Histogram equalization ideally distributes the contrast of the image using the image's histogram. An image can be represented as a data array in the form of:

$$X = \{X(i,j) \mid X(i,j) \in \{X_0, X_1, \ldots, X_{L-1}\}\}. \quad (3)$$

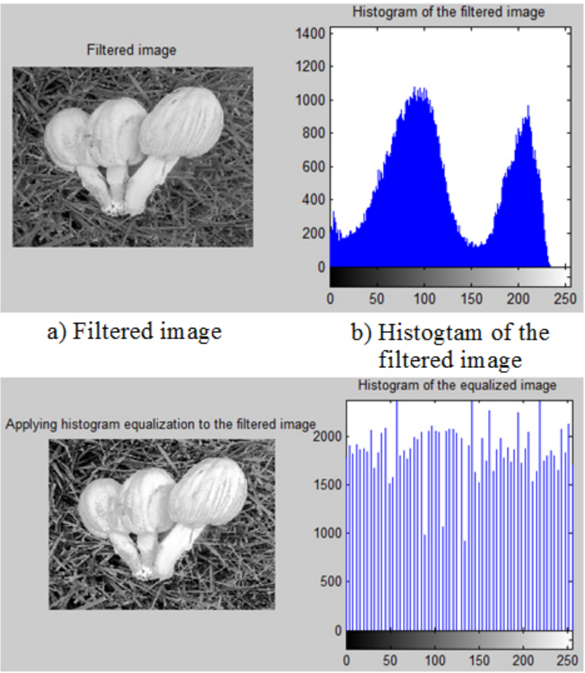
In this data array, every component can be composed of L intensity level. In $X(i,j)$ image plane, (i,j) represents normalized intensity of a pixel. X_k is kth intensity level. Equation (4) is used to obtain probability distribution function (PDF) of the image [5]:

$$p(X_k) = \frac{n_k}{n}, 0 \leq X_k \leq 1, \quad (4)$$

where n is the total number of pixels in the input image and n_k is the number of X_k in the image X. To obtain a better contrast image Eq. (5) is used:

$$s_k = \sum_{j=0}^{L-1} p(X_j) = \sum_{j=0}^{L-1} \frac{n_j}{n} = 1, \quad (5)$$

where L is the total number of possible grey levels (such as 255 for 8 bit depth), s_k is the grey conversion value for a better contrast image. Figure 4(a) shows filtered image, Fig. 4(b) shows histogram image, Fig. 4(c) shows histogram equalized image and Fig. 4(d) shows the histogram of the image. As it can be seen from the figures after histogram equalization, intensities are equally distributed according to the pixels.



a) Filtered image

b) Histogtam of the filtered image

c) Applying histogram equalization to the filtered image

d)Histogram of the equalized image

Fig. 4: Histogram equalization process.

2.2. Segmentation

In this step segmentation process will be explained.

1) K-Means Method

K-Means is one of the most widely used uncontrolled learning processes. This method ensures that all data belong to a single cluster. This provides an efficient clustering mechanism. K-Means algorithm groups n data points into C number of clusters. Goal is at the end to have a high level of similarity in the clusters and low level of similarity among the clusters [1], [6].

Squared error criterion E is widely used to obtain the distance of cluster members to the cluster center. For the most successful clustering, E value is expected to be small. Equation (6) is used for obtaining the sum of the squares of the distances of members to the cluster center:

$$E = \sum_{i=1}^{C} \sum_{k=1}^{N} dist^2\left(x_k, s_i\right). \tag{6}$$

At the end of clustering N points are divided into C clusters. For the distance calculation Euclid equation given in Eq. (7) is commonly used [6]:

$$dist^2\left(x_k, s_i\right) = \left\| x_k^{(i)} - s_i \right\|^2. \tag{7}$$

2) Color Segmentation with Modified K-Means Algorithm

Figure 5 shows the GUI structure used for mushroom segmentation and finding cap width. On the GUI shown in Fig. 5, mushroom image to be processed, grayscale k-means analysis results, color k-means analysis results, k-means clustering image is provided. Also from the GUI, edges of the interested segmentation are determined and cap width of the segmented mushroom is calculated.

Algorithm 1 provides software algorithm that is used for Gray K-Means segmentation. This stage is composed of 9 steps:

- **Preprocessing**: At this step previously described filtering and histogram equalization processes are performed. Lines 1–4 correspond to these processes in the algorithm.
- **Determination of the cluster center:** In the study, number of cluster is determined to be 3. Because of that 3 intensity levels chosen from HisteqMushroom image is set as cluster starting point. Line 5 of the algorithm shows this process, c_1, c_2 and c_3 keeps the center values.
- **Calculating distance from the cluster center:** At this step distance between each pixel in the image and the cluster centers c_1, c_2 and c_3 is calculated. Codes of the distance function are given in Algorithm-2. Function calculates the distances and keeps them in distance1, distance2, and distance3 variables.
- **Calculation of clustering data sums and producing K-Means cluster map:** Step 11 through 30 of Algorithm-1 reflects these steps. At the end of the process graykmeans variable holds the gray level k-means clustering map. `c1_sum` has cluster c_1, `c2_sum` has cluster c_2 and `c3_sum` has cluster c_3 data sum and `c1count`, `c2count`, `c3count` variables contain the respective cluster's number of members. Coordinates that lie inside cluster 1 are painted in black, coordinates that lie inside cluster 2 are painted in gray and coordinates that lie inside cluster 3 are painted in white.

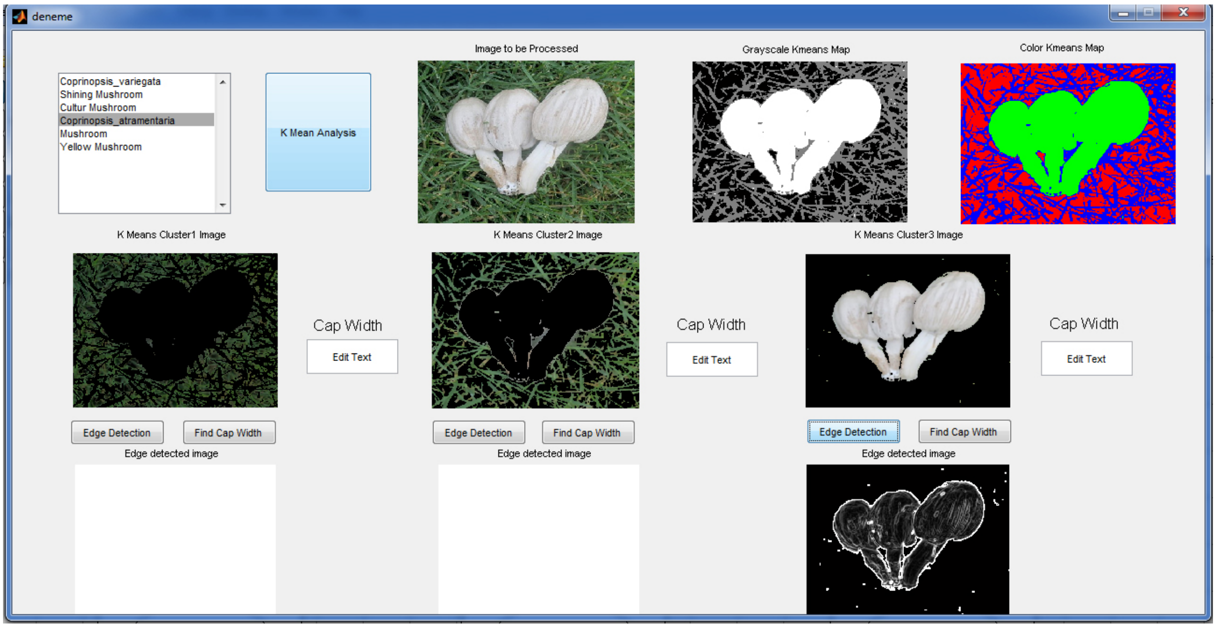


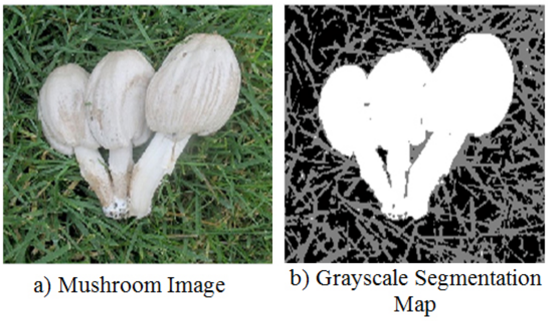
Fig. 5: GUI structure of the software that segments mushroom and calculates cap width.

- **Stopping Iteration:** `graykmean` variable and `comp_kmean` variable are equalized in line 10. If `comp_kmean` and `graykmean` are equal, iteration is stopped. Lines 31, 32 and 33 reflects this process.

- **Calculating the cluster centers:** Lines 35, 36, and 37 reflects this process. By dividing `c1_sum`, `c2_sum` and `c3_sum` into `c1count`, `c2count` and `c3count` respectively new cluster centers (c_1, c_2 and c_3) are obtained. Figure 6b) shows the grayscale segmentation map image that is stored in grayscale variable after running Algorithm-1.

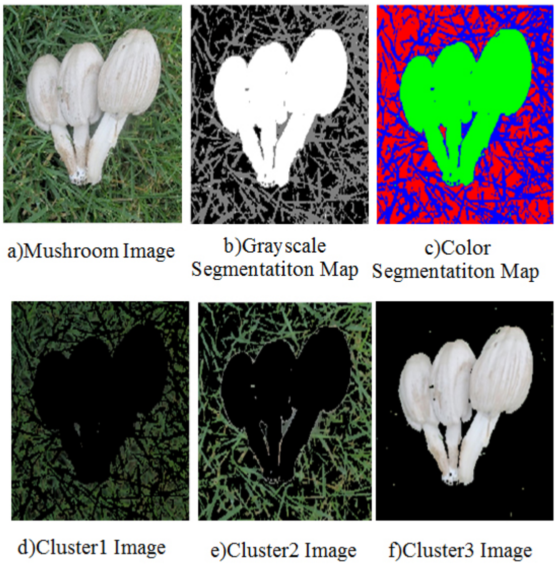
- **Obtaining color segmentation map:** Grayscale segmentation map obtained in Algorithm-1 is converted into color segmentation map by Algorithm-3. Algorithm-3 turns black colored coordinates into red and corresponding data are moved to cluster 1. Similarly gray colored coordinates turned into blue and corresponding data are moved to cluster 2 and white colored coordinates turned into green and corresponding data are moved to cluster 3. Figure 7 show Color Segmentation, Cluster1 Image, Cluster2 Image ve Cluster3 Images obtained after running algorithm 3. As it can be observed in Fig. 7f) mushroom image is successfully segmented.

- **Determining the cluster image boundaries:** At this stage sobel filter is applied to the user selected cluster image. Edge Detection button controls the boundary drawing action. After clicking


a) Mushroom Image

b) Grayscale Segmentation Map

Fig. 6: Grayscale Segmentation Map.


a)Mushroom Image

b)Grayscale Segmentatiton Map

c)Color Segmentatiton Map

d)Cluster1 Image

e)Cluster2 Image

f)Cluster3 Image

Fig. 7: Color segmentation map and cluster images.

the button Edge Detected Image shows the filtered image.

- **Calculating cap width:** Find Hat Width button controls the action. `graykmean` variable is used in cap width finding algorithm. When the button is clicked by using `graykmean` value and gray level color information image is scanned from the top row to the bottom and cap width is calculated in pixels. Pixel value is then multiplied with preset calibration parameter (CP) to find the cap width in cm. In order for cap width to be measured by the software image must contain only one mushroom. On the test image cap width of the mushroom seen in cluster 3 image is calculated as 15.32 cm (Fig. 9).

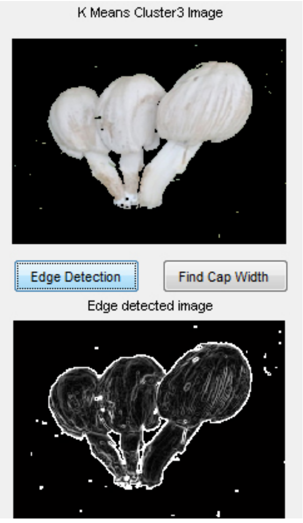
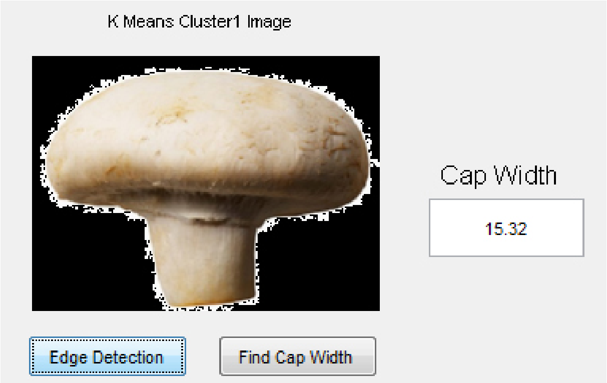


Fig. 8: Applying sobel filter to Cluster3 Image.



Cap Width

15.32

Fig. 9: Cap width calculation of cluster 1 image.

2.3. Showing the Results

Developed software shows the analysis results in separate image boxes. Calculated cap width value is provided in the textbox.

Algorithm 1 Gray K-Means segmentation algorithm.

```
MushroomImage = imread('Mushroom.jpg')
AverageMushroom = imfilter(MushroomImage, AverageFilter,'replicate');
HisteqMushroom= histeq(AverageMushroom)
[height width] = size(HisteqMushroom)
c1,c2 and c3 = HisteqMushroom (randi(height,1,1), randi(width,1,1));
continue = 1;
while continue = 1 do
  c1_sum = 0; c2_sum = 0; c3_sum = 0;
  c1count = 0; c2count = 0; c3count = 0;
  comp_kmean == graykmean
  for i = 1:Height do
    for j = 1:Width do
      distance1 = distance(HisteqMushroom(i,j), c1);
      distance2 = distance(HisteqMushroom(i,j), c2);
      distance3 = distance(HisteqMushroom(i,j), c3);
      if distance1 < distance2 && distance1 < distance3 then
        graykmean(i,j) = 0;
        c1_sum = HisteqMushroom (i,j) + c1_sum;
        c1count = c1count + 1;
      else if distance1 > distance2 & distance2 < distance3 then
        graykmean (i,j) = 0.5;
        c2_sum = HisteqMushroom (i,j )+ c2_sum;
        c2count = c2count + 1;
      else if distance1 > distance3 && distance2 > distance3 then
        graykmean (i,j) = 1;
        c3_sum = HisteqMushroom (i,j) + c3_sum;
        c3count = c3count + 1;
      end if
    end for
  end for
  c1 = c1_sum/c1count;
  c2 = c2_sum/c2count;
  c3 = c3_sum/c3count;
end while
```

Algorithm 2 Distance Calculation.

```
function Euclid = distance (v1,v2)
Euclid =((v1−v2)^2)^(1/2); (=ABS(v1−v2))
end function
```

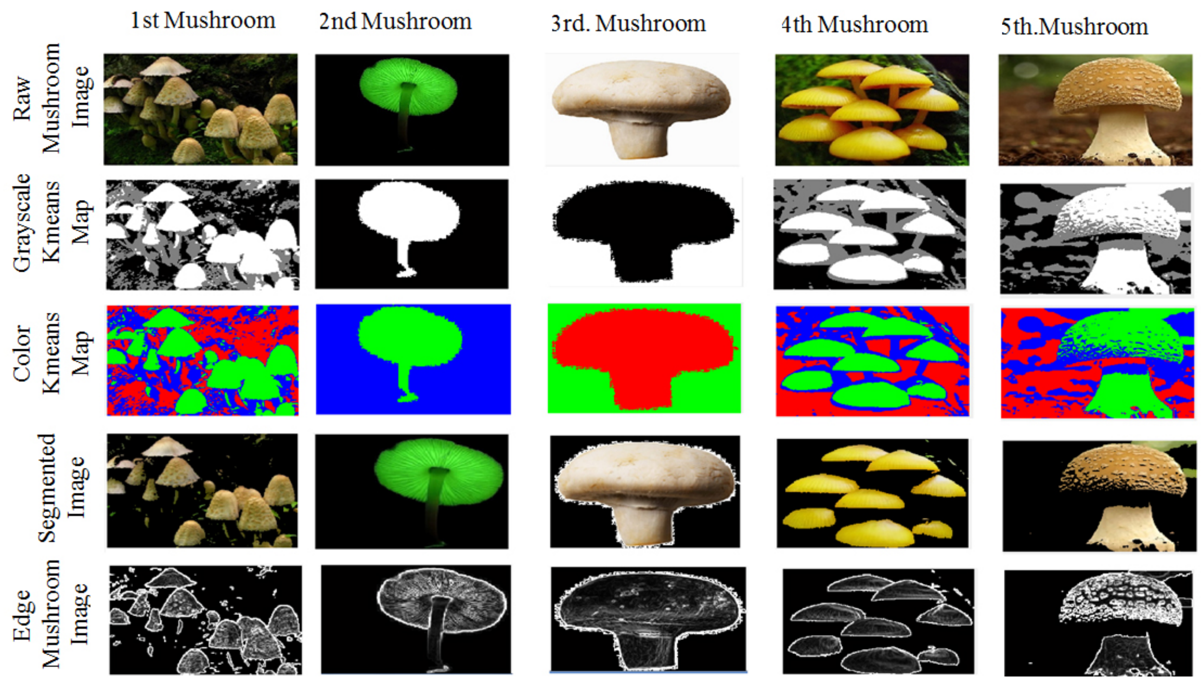


Fig. 10: Segmentation performance test images.

Algorithm 3

```
for i = 1:Height do
    for j = 1:Width do
        if graykmean(i,j) == 0 then
            cluster1(i,j,:) = MushroomImage(i,j,:);
            kcolormean(i,j,1) = 255;
            kcolormean(i,j,2) = 0;
            kcolormean(i,j,3) = 0;
        else if graykmean (i,j) == 0.5 then
            cluster3(i,j,:) = MushroomImage(i,j,:);
            kcolormean(i,j,1)=0;
            kcolormean(i,j,2)=255;
            kcolormean(i,j,3)=0;
        end if
    end for
end for
```

3. Experimental Results

Several tests performed with the software whose GUI structure is shown in Fig. 5. In the first phase segmentation performance tests are carried out and in the second phase cap width measurement performance tests are carried out.

3.1. Segmentation Performance Test

At this phase segmentation performance of the designed system is tested. Five different mushroom images are used for the tests. Test results are provided in Fig. 10. First row of Fig. 10 shows the images to be processed. Second row contains grayscale k-means maps, and third row contains color k-means maps of the corresponding images. In the fourth row segmented mushroom images can be observed. Fifth row contains edge determined segmented mushroom images. Results show that the designed system successfully segments the provided images. System performance does not change in complex background images.

3.2. Mushroom Cap width Measurement Performance

In this step Cap Width Measurement is performed. Our software measured the cap width of the mushroom as 8.48 cm but the real cap width of the mushroom is 8.66 cm. To find the measurement error in this result, Absolute Error (Δae) and Relative Error ($re\%$) variables in Eq. (8) and Eq. (9) is used. Error analysis resulted in following error values: $\Delta ae = 0.18$ cm, $re\% = 2.12$:

$$\Delta ae = \left| m_{\text{measured}} - m_{\text{real value}} \right| , \tag{8}$$

$$re\% = \frac{|\Delta m|}{m_{\text{real value}}} \cdot 100, \tag{9}$$

$$\Delta ae = |8.48 - 8.66| = 0.18 \text{ cm}, \tag{10}$$

$$re\% = \frac{0.18}{8.66} \cdot 100 = 2.12. \tag{11}$$

During the test 19 different mushrooms with different size and type are used. Figure 11 provides the results ordered ascending according to $m_{\text{real value}}$.

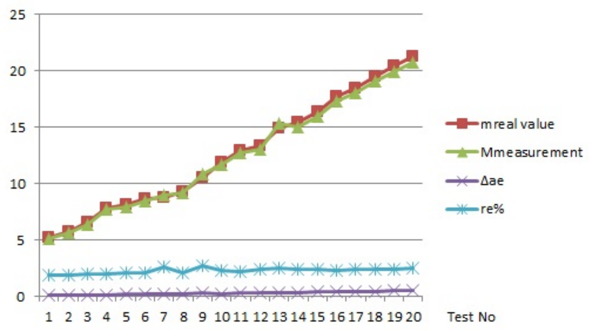


Fig. 11: Test result graphics.

Statistical analysis is performed on Δae and $re\%$ values found during tests and results are provided in Tab. 1. As results show Δae and $re\%$ are very low.

Tab. 1: Statistical results.

Statistical results of Δae [cm]		
Max	Arithmetic mean	Standard deviation
0.530	0.296	0.134
Statistical results of $re\%$ [cm]		
Max	Arithmetic mean	Standard deviation
2.756	2.286	0.231

4. Conclusion

In this study we developed a GUI based software for K-Means image segmentation. Raw input image is filtered and histogram equalized at the beginning of the process. On the processed image segmentation is performed with k-means method. To improve segmentation performance k-means algorithm is improved with modifications. Test results show that designed software successfully performs segmentation even on complex background images. The analysis also shows that histogram processes and noise reduction processes play an important role in a successful segmentation process. Software also has the capability to measure cap width of a mushroom to provide information about the size of the mushroom in the image.

References

[1] YAO, H., Q. DUAN, D. LI and J. WANG. An improved k-means clustering algorithm for fish image segmentation. *Mathematical and Computer Modelling*. 2013, vol. 58, iss. 3–4, pp. 790–798. ISSN 0895-7177. DOI: 10.1016/j.mcm.2012.12.025.

[2] ZHANG, Y., D. HUANG, M. JI and F. XIE. Image segmentation using PSO and PCM with Mahalanobis distance. *Expert Systems with Applications*. 2011, vol. 38, iss. 7, pp. 9036–9040. ISSN 0957-4174. DOI: 10.1016/j.eswa.2011.01.041.

[3] LAO, Z., D. SHEN, D. LIU, A. F. JAWAD, E. R. MELHEM, L. J. LAUNER, R. N. BRYAN and Ch. DAVATZIKOS. Computer-Assisted Segmentation of White Matter Lesions in 3D MR Images Using Support Vector Machine. *Academic Radiology*. 2008, vol. 15, iss. 3, pp. 300–313. ISSN 1076-6332. DOI: 10.1016/j.acra.2007.10.012.

[4] GONZALEZ, R. C. and R. E. WOODS. *Digital image processing*. Upper Saddle River: Prentice Hall, 2002. ISBN 978-02-011-8075-6.

[5] YOON, H., Y. HAN and H. HAHN. Image Contrast Enhancement based Sub-histogram Equalization Technique without Over-equalization Noise. *World Academy of Science, Engineering and Technology*. 2009, vol. 3, iss. 6, pp. 176–182. ISSN 2010-3778.

[6] HAN, J., M. KAMBER and A. K. H. TUNG. Spatial Clustering Methods in Data Mining: A Survey. In: *Geographic Data Mining and Knowledge Discovery*. New York: Taylor, 2001, pp. 1–29. ISBN 04-152-3369-0.

About Authors

Eser SERT received the M.Sc. degree in 2010 and the Ph.D. degree in 2013, all in Computer Engineering at Trakya University, Turkey. Currently he is Assistant Professor at Computer Engineering of Kahramanmaras Sutcu Imam University. His research interests include 3D modeling system, image processing, computer vision, field programmable gate arrays (FPGA) and artificial inteligence.

Ibrahim Taner OKUMUS was born in Kahramanmaras, Turkey. He received his M.Sc. and Ph.D. degrees from Sysracuse University, USA. He is currently working as an Associate Professor at Computer Engineering Department of Kahramanmaras Sutcu Imam University, Turkey. His research interests include wireless networks, software defined networks, wireless sensor networks and computer networks in general.