

Grape Leaves Segmentation using an Improved Graph-Based Approach

Vasudevan Narasimman*, Karthick Thiyagarajan

Department of Data Science and Business Systems, School of Computing, SRM Institute of Science and Technology, Kattankulathur, Tamilnadu, 603203, India

Corresponding Author Email: vn8049@srmist.edu.in



<https://doi.org/10.18280/ria.360517>

ABSTRACT

Received: 19 October 2022

Accepted: 6 December 2022

Keywords:

segmentation, leaf diseases, pixel accuracy, feature extraction, classification

The study of plant pictures is very beneficial to agriculture and is sustainable. It is used to precisely and often record data on plant development, yield, breadth, and height of the plant, leaf area, etc. The quantity of leaves in the plants directly affects plant development, which is one of the most important characteristics to be examined among these plant characteristics. To extract leaves from photos of grape plants, a novel technique known as an enhanced graph-based approach is proposed in this study. The suggested procedure comprises two phases. Red, Green, and Blue (RGB) to Hue, Saturation, and Value (HSV) conversion and background removal are required in the initial phase for picture improvement. In the second stage, a novel graph-based technique and the Circular Hough Transform (CHT) are used to extract the leaf area from the plant picture. Theni District, Tamilnadu, India's grape leaf real-time datasets have been used in the experimentation for the suggested study. The segmentation pixel accuracy of the suggested approach is 92.4%, and the Mean Intersection over Union (MIoU) value is 86.2%, which is superior to the methods currently in use.

1. INTRODUCTION

The grape has been one of the most significant fruit crops from an economic standpoint since it is one of the most lucrative fruits for Indian farmers to grow. It is also one of the most well-known fruits not only in India but all over the globe because it is used to make raisins and wine. The nutritional value of grapes and their use as natural treatments for a variety of health problems also contribute to their relevance. Additionally, it has been lucrative due to its broad usage and ease of growth in a variety of climatic conditions.

Infectious agents (viral, bacterial, and fungal) and non-contaminating factors may cause plant infections. Numerous illnesses are spread via plant leaves. Plant illnesses known as biotic diseases are brought on by infectious pathogens. Abiotic illness indicates that there are no contaminating compounds present in plants. Abiotic infections are not as much of harmful due to not being capable of being passed on. Therefore, abiotic illnesses are most at risk. Rust, mildew spots, canker, and birch are the maximum prevalent diseases reviewed in the testing of recognition systems on plants. The most severe infections in plants are caused by fungi. By damaging and highlighting the plants' cells, the fungus contaminates them. The diseases include a list of fungal diseases like mites, scorch, scab, rust, and blight. Plants that are bacterially ill display signs like leaf spots and wilts. These infections arise at an advanced stage and are very difficult to identify. Bacterial illnesses include wilting, bacterial spots, etc. The ones that harm plant leaves are the most difficult to diagnose. The viruses are undetectable and won't be quickly found till later. Lack of nutrition and chemicals often disrupt viral infections. The viral illnesses include abdication, leaf curl, and mosaic among others. Image handling methods aim to increase agricultural yield via crop

monitoring. Computer visualization and artificial approaches have been used in several research projects on the detection of plant diseases [1]. This study article's goal is to quickly review the investigation articles that used aspects of machine learning and image processing. Pictures acquisition, image preprocessing [2], image segmentation, feature abstraction, and recognition of plant disease [3] are generally used in image processing methodologies and machine learning. The primary procedures that are used by many studies to accomplish automated leaf disease detection are listed below. In common, image processing [4] uses phases to address the issue, namely classification.

Image Acquisition: The leaf pictures are photographed and saved, which is the first and most fundamental stage in image processing [5].

Pre-Processing: De-noising and picture smoothing are done in this part. Here, RGB photos are transformed into greyscale images, and the contrast of the images is added [6].

Segmentation: The picture is divided into distinct segments during this phase, which is crucial for identifying the required feature during image analysis [7].

Feature Extraction: Important features like form, texture, and color are obtained, which reduces the number of resources needed to offer the description [8].

Classification: According to the features recovered, the different types of leaf diseases are categorized, and these classifications indicate to which group a certain leaf belongs [9].

One of the most important crops is grapes, especially because winemaking requires grapes of the highest quality. Unfortunately, illness causes the quality to decline. Additionally, several researchers have attempted to create an automated approach to identify the illness. The segmentation

of leaf pictures and their structure estimate [10] was accomplished using the author's suggested parametric active polygon model (PAPM). In the study, a light polygonal framework was used as the model to describe the leaf structure and fit the leaf picture. The framework's development included both leaf segmentation and structural representation. However, if the improvement increased, it would be necessary to boost precision. Tian et al. [11] looked at cucumber illnesses and detection techniques. In this study, they describe many filtering applications, including range segmentation using colors to extract color information from the lesion site and gray-level co-occurrence matrix quality factors for texture and structure. By measuring and calculating the distance, leaf disease may be identified. In any case, this strategy is simple in terms of computation complexity [12]. To properly analyze pictures of lamina using shape analysis, the author presented the lamina2shape algorithm (L2SA). This technique was tested and analyzed using winter wheat. Images of winter wheat were first acquired by the flatbed scanner. After that, the threshold for the green, blue, and red channel values of the image's pixel ratio set against the absence of a colored background allowed for the identification of the lamina. It is necessary to compute the numerical lamina function to determine the lamina's picture. The lamina, nevertheless, was not finished well. Larese et al. [13] suggested using leaf vein image (LVI) characteristics to categorize legumes using an artificial segmentation and classification approach. Manly species of legumes such as white beans, soybeans, and red beans were developed. Initially, a conventional scanner is used to capture a picture of the leaf. Unrestricted adaptive thresholding methods (ATT) and hit-or-miss transform are offered for segmentation performance. Various classifiers use various morphological processes. In any case, the intricacy was rather significant. Shantkumari et al. [14] segments illumination-invariant data to research a grape picture using an adaptive snake model. Utilizing several types of methods, the vegetation was extracted from the plant in the photograph. Following that, absolute segmentation and common segmentation were addressed, and their effectiveness was measured against that of several alternative techniques. The precision was much enhanced. Dey et al. [15] provided an adaptive thresholding method for the context-dependent segmentation approach used by the Canny edge detector and global adaptive binarization threshold image segmentation to randomly select segments from a live network system. And also the enhanced convolutional-deconvolutional network-based segmentation, dense deconvolution network-based segmentation, and Edge gradient and long-distance dependency-based segmentation were used for effective image segmentation.

Here, we have outlined a graph-based technique for identifying and segmenting likely regions. This research effort makes the following contributions.

1. The graph-based approach for area identification and segmentation is proposed in this study effort.
2. Two phases make up the graph-based algorithm. Image improvement comes first, followed by leaf extraction.
3. Image enhancement is accomplished by the conversion of RGB to HSV.
4. The graph construction technique is used to extract the leaf area following image enhancement.
5. MIoU and pixel accuracy are employed to assess the effectiveness of this work.

2. METHODS AND MATERIALS

2.1 Proposed method

The graph-based approach that has been suggested in this study is built to efficiently extract leaves from the provided plant photos. In Figure 1, the suggested technique is shown. Image enhancement and leaf segmentation make up the two processes. The statistical model for removing the lighting effect and the graph model for extracting the leaves have both been utilized in the suggested technique.

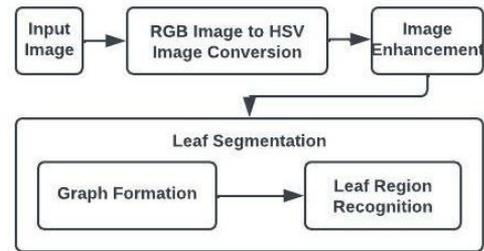


Figure 1. Process of extracting the leaves

2.1.1 RGB image to HSV image conversion

RGB lights are put organized in different ways to create a wide range of colors in the additive RGB color model. The first letters of the 3 basic colors red, green, and blue were used to make the model's name. Artists typically utilize the HSV color theory since it is frequently more real to think about a color in terms of saturation and hue. The components and colorimetry of HSV, which is a conversion of an RGB colorspace, are related to the RGB colorspace from which it was formed [16]. In the suggested work, the plant's RGB picture is first transformed into an HSV image. Since the V plane of an HSV picture correlates to the brightness of the image, image enhancement is done there.

The following equations are used to translate an image's RGB values into HSV values.

$$H = \begin{cases} H_1, & \text{if } B \leq G, \\ 360^\circ - H_1, & \text{if } B > G, \end{cases} \quad (1)$$

where,

$$H_1 = \cos^{-1} \left\{ \frac{0.5[(R - G) + (R - B)]}{\sqrt{(R - G)^2 + (R - B)(G - B)}} \right\} \quad (2)$$

$$S = \frac{m(R, G, B) - m(R, G, B)}{m(R, G, B)} \quad (3)$$

$$V = \frac{m(R, G, B)}{255} \quad (4)$$

2.1.2 Image enhancement

By transforming the input picture into an HSV picture, the color component and luminance factor are separated. Only the value element of the supplied picture is enhanced, leaving the hue and saturation components alone. Contrast enhancement and luminance enhancement are two distinct procedures that make up the value enhancement component.

The luminance enhancement is used to the input image's value component with the help of a nonlinear transfer function that was specifically created for the task and is

described in the study [17]. Let $V_1(x, y)$ stand for the HSV-normalized V channel, and let $V_2(x, y)$ represent the transmitted value after using the nonlinear transfer function provided here.

$$V_2 = \frac{V_1^{(0.75z+0.25)} + 0.4(1-z)(1-V_1) + V_1^{(2-k)}}{2} \quad (5)$$

where, z is the image-dependent parameter and is defined as follows.

$$k = \begin{cases} 0 & \text{for } h \leq 50 \\ \frac{h-50}{100} & \text{for } 50 < h \leq 150 \\ 1 & \text{for } h > 150 \end{cases} \quad (6)$$

In the example above, h is the value level (V) that corresponds to the cumulative probability distribution function of 0.1. The degree of brightness amplification for each pixel value is defined by the parameter k in Eq. (5). Most photos don't have uniform lighting throughout. As a result, the picture features cannot be preserved while using the transfer function by assessing parameter k worldwide. We divide the image into small blocks of equal size and compute the parameter k for each block. We further break the blocks into subblocks to alleviate the blocking artifacts and region transition issues. We next interpolate the formerly obtained k for each block to get the value k for each subblock. The parameter k is interpolated from blocks to subblocks using the bilinear interpolation approach. We utilize the parameter value for k as the center coordinate value for the matched block. All of the subblocks within the center coordinate position of the bordering blocks are interpolated using the four-parameter k values from the bordering blocks. Now, depending on the value of k for each sub-block, the form of the transfer function varies.

To increase the overall image's quality, the contrast information from the original image must be boosted by utilizing the contrast improvement process. This method uses Gaussian convolution using function $G(x, y)$ on the V channel of the input image in HSV space. The convolution may be written as follows.

$$V_3(x, y) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} V(m, n)G(m+x, n+y) \quad (7)$$

The brightness data from the neighboring pixels is included in the convolution result, as demonstrated by V_3 in Eq. (7). The output of the Gaussian convolution is now compared to the value of the center pixel to ascertain the level of contrast enhancement. Eq. (8) below describes this procedure.

$$V_4(x, y) = 255 V_2(x, y)^{E(x, y)} \quad (8)$$

where,

$$E(x, y) = \left[\frac{V_3(x, y)}{V(x, y)} \right]^g \quad (9)$$

where, g is the variable used to fine-tune the contrast enhancement procedure based on the original value component picture in HSV space. The following equation is used to calculate this parameter g .

$$g = \begin{cases} 1.75 & \text{for } \sigma \leq 2 \\ \frac{27-2\sigma}{13} & \text{for } 2 < \sigma < 10 \\ 0.5 & \text{for } \sigma \geq 10 \end{cases} \quad (10)$$

where, σ stands for the original value component image's each block's standard deviation. In the first use of this contrast improvement approach, the standard deviation is found worldwide. A linear connection between g and σ is shown by Eq. (10). Again, the same value of g for boosting the contrast for each input pixel is inappropriate in this case owing to variable lighting in various regions of the input image. In rare instances, the visual information may be inaccurate. By breaking the input picture's value channel into smaller blocks in HSV space and determining the parameter g for the image's each block separately, we may solve this issue using the same method as previously explained. Multi-scale convolution is used for greater performance. The closest neighbor pixel brightness information is taken into account if the small-scale Gaussian function is used for convolution with the input picture. On the other hand, the whole brightness information is taken into account if we use a large-scale Gaussian function. The final average picture is calculated from the gathered scale-convoluted photos. However, the multi-scale convolution lengthens the calculation time and makes it more difficult. The "detail" and "amount" tools are used in the local dynamic range compression processing to translate a more dynamic picture to a lower level. The value component image's ultimate improvement is provided by Eq. (8). As illustrated in Figure 2, the improved color picture is finally created in RGB space by fusing the improved value channel with the hue and saturation channels in HSV space. This picture-enhancing process leaves the saturation and hue channels alone.

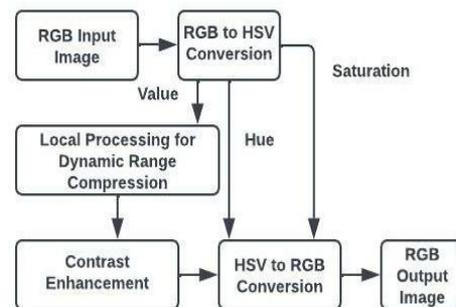


Figure 2. Image enhancement

2.2 Leaf segmentation

We use a graph-based method for segmentation. Assume that $G = (V, E)$ is an un-directed graph with edges (v_i, v_j) corresponding to pairs of nearby nodes or vertices and nodes $v_i \in V$, representing the collection of items to be separated. A weight $w(v_i, v_j)$ equivalent to each edge $(v_i, v_j) \in E$ exists as a nonnegative indicator of the difference between adjacent components v_i and v_j . Pixels make up the components of V when segmenting a picture, and the weight of each edge indicates how unlike the two pixels connected by that edge are from one another. The formulation presented here, however, is not reliant on these explanations. A segmentation S in the graph-based method is the division of a variable (V) into elements, where each element or component (or area) $C \in S$ relates to a linked element in a graph $G1 = (V, E1)$, where $E1$

∈ E. In other confrontations, a portion of the edges in E will cause any segmentation. The effectiveness of segmentation may be evaluated in a variety of ways, but generally speaking, we need the parts within the element to be similar to one another and the elements inside separate components to be unlike. Accordingly, edges connecting vertices within the same element must have relatively low weights, but edges connecting nodes inside separate elements should have greater weights. For two provided areas, let's say C1 and C2, a decent pairwise region comparison is required to assess the segmentation's quality. The following equation served as its description.

$$D(C_1, C_2) = \begin{cases} \text{True} & \text{if Dif}(C_1, C_2) > \text{MInt}(C_1, C_2) \\ \text{False} & \text{otherwise} \end{cases} \quad (11)$$

The least-weight edge that connects a node v_i in component C1 to a node v_j in component C2 distinguishes the two components. This is the internal variation between the C1 and C2 components that are controlled by the k-factor. The factor k in the definition of minimum internal difference (MInt) serves to differentiate between the maximum internal difference and MInt. As $r(C)$ establishes the minimum distance by which components must vary from the internal nodes of a component.

$$r(C) = \frac{k}{|C|} \quad (12)$$

here, the properties of k are as follows.

- If k is big, it makes people want bigger things.
- k doesn't specify a minimum size for components

2.2.1 Leaf area recognition

The graph for each photo is created using the graph-based technique that was discussed in the section that came before this one. Every pixel that makes up an image is regarded as a vertex in this situation. The source vertex served as the starting point for the graph's construction. The pairwise area comparison is used to connect the source vertex and its neighboring vertices, and this process is repeated to build the graphs. Here is a presentation of the graph's building process.

Algorithm 1: Graph-Based Leaf Extraction

- Step1: Read the image
- Step2: Consider each pixel of an image as vertices. And choose the source pixel or vertex.
- Step3: Construct the graph by combining the source vertex with the nearby vertices if these vertices have similar properties by pairwise region comparison.
- Step4: Use the threshold value $r(C)$ to differentiate or disconnect the graph.
- Step5: Repeat step3 and step4 to form the graphs for the whole image.
- Step6: Finally by using circular hough transform, identify the leaves from the segmented graph.

We will be in charge of locating the leaves-shaped graph that is still there in the region where the leaves were obtained after the graph has been constructed. Because leaves are often spherical, the Circular Hough Transform [18] is used to identify them. While concentrating on this leaf area, the Circular Hough Transform is used. The circular region on the surface of each leaf is used to identify it. The plant depiction in the center of the circle displays the whole leaf surface area.

3. DATASET AND EXPERIMENTAL DESIGN

The real-time group of grape leaves dataset was used for this suggested job. The datasets were taken with a Canon SD1000 camera in the Theni district of Tamil Nadu, India, and are utilized for the segmentation work that is being suggested. For three weeks, images for the dataset were captured once every six hours. 1110 (512,512 pixels) photographs with animated backdrops make up the piece. The list of example images is in Figure 3. The processing algorithm was built in the MATLAB 2011a environment, and the experiment's picture size was 512 pixels by 512 pixels. For picture enhancement, RGB to HSV conversion, graph generation, and precise object recognition, MATLAB provides a wide range of functions and packages.



Figure 3. Grape leaves images

4. EXPERIMENTAL RESULTS AND DISCUSSIONS

The segmentation of the leaf area in the suggested approach is done using a graph-based algorithm, which not only improves segmentation accuracy but also preserves the technique's resilience to shadows and reflections. The image's number of rows and columns is used to assess how well the suggested technique for separating the leaves works. For ease, the removal of the leaves is divided into two parts. The first phase is image improvement, next comes leaf section location.

The plant's RGB image will first be transformed into an HSV image in the proposed work. Since the brightness of the image coincides with the HSV picture's V Plane, image enrichment is done on this plane. A visual depiction of the enhanced image is shown in Figure 4 In this illustration, the RGB to HSV color space conversion and removal of the background enhance the original image.

Each pixel that makes up an image is regarded as a vertex in this study. The source vertex served as the starting point for the graph's construction. The pairwise area comparison is used to connect the source vertex and its neighboring vertices, and this process is repeated to build the graphs. We will be in charge of locating the leaves-shaped graph that is still there in

the region where the leaves were obtained after the graph has been constructed.

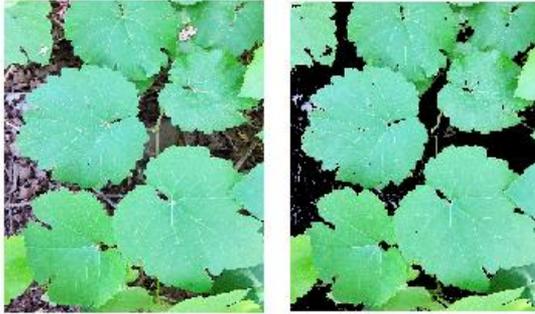
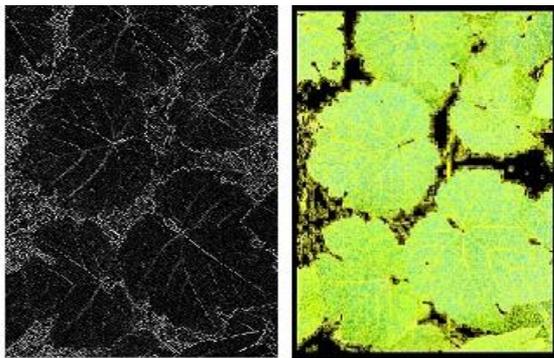


Figure 4. Enhanced image



(a) Graph Construction (b) Leaf Identification

Figure 5. Leaf identification

Because leaves are often spherical, the CHT is applied to identify them. When concentrating on this leaf area, the CHT is used. The circular region on the surface of each leaf is used to identify it. The plant depiction in the center of the circle displays the whole leaf surface area. Figure 5 and Figure 6 show the identified and segmented grape leaves, respectively.

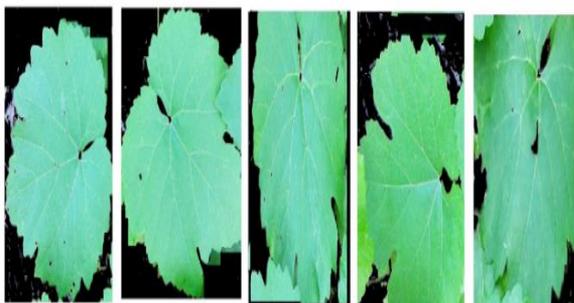


Figure 6. Segmented grape leaf images

Pixel accuracy and MIoU are the two most often used measures for determining how well an image segmentation algorithm performs. Although pixel accuracy is very easy to accomplish, classes that dominate the scene severely distort it. The easiest way to assess how well an image segmentation model works is to look at its pixel accuracy. We must first determine the True Negative(TN), True Positive(TP), False Negative(FN), and False Positive(FP) values to proceed. The formula for calculating pixel accuracy is

$$\text{Pixel Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (13)$$

- **True Positive:** pixel considered properly as X
- **False Positive:** pixel considered improperly as X
- **True Negative:** pixel considered properly as not X
- **False Negative:** pixel considered improperly as not X

Another way to assess the predictions made by an image segmentation model is to use intersection over the union. Through the intersection and union of the Ground Truth and the Prediction, this technique determines performance. The following formula may be used to get the IoU value.

$$IoU = \frac{\text{Intersection}}{\text{Union}} = \frac{TP}{TP + FP + FN} \quad (14)$$

The recommended technique, known as "graph-based grape leaf segmentation," is compared with the pixel accuracy and MIoU values of the various algorithms. Additionally, Table 1 result shows that the suggested approach has higher pixel accuracy and MIoU values than the enhanced convolutional-deconvolutional network-based segmentation [19], dense deconvolution network-based segmentation [20], and Edge gradient and long-distance dependency-based segmentation [21].

Table 1. Segmentation comparisons

Methods	Pixel Accuracy	MIoU
Enhanced convolutional-deconvolutional network-based segmentation	78.6%	74.9%
Edge gradient and long-distance dependency-based segmentation	89.1%	82.3%
Dense deconvolution network-based segmentation	86.2%	79.1%
Graph-Based Segmentation (Proposed)	92.4%	86.2%

5. CONCLUSION

There are now stricter constraints for computation cost, model size, and segmentation accuracy in picture segmentation due to the ongoing progress in automated driving and security monitoring. This is why a better graph-based approach to the grape leaf image segmentation model is suggested. Graph generation and picture enhancement are used to enhance the graph-based segmentation model. For grape leaf segmentation in complicated settings, the graph construction was additionally merged with the circular hough transform. Using real-time datasets, the suggested model is empirically proven. Results show that the datasets of the proposed method have greater pixel accuracy and MIoU values than existing comparison techniques. The approach that is suggested in this research guarantees both accurate segmentation and operational efficiency. The performance of the semantic segmentation model will be further enhanced by our efforts to increase target boundary segmentation accuracy, our capacity to effectively segment tiny targets, and our ability to solve the discontinuous target segmentation issue.

ACKNOWLEDGMENT

The authors are grateful to the supervisor for his direction and support during this research.

REFERENCES

- [1] Shrivastava, G. (2021). Review on emerging trends in detection of plant diseases using image processing with machine learning. *International Journal of Computer Applications*, 174(11): 39-48. <https://doi.org/10.5120/ijca2021920990>
- [2] Berthet, A., Dugelay, J.L. (2020). A review of data preprocessing modules in digital image forensics methods using deep learning. *IEEE International Conference on Visual Communications and Image Processing*, pp. 281-284. <https://doi.org/10.1109/VCIP49819.2020.9301880>
- [3] Liu, X., Min, W., Mei, S., Wang, L., Jiang, S. (2021). Plant disease recognition: a large-scale benchmark dataset and a visual region and loss reweighting approach. *IEEE Transactions on Image Processing*, 30(1): 2003-2015. <https://doi.org/10.1109/TIP.2021.3049334>
- [4] Vasudevan, N., Karthick, T. (2022). Analysis of plant leaf diseases recognition using image processing with machine learning techniques. *IEEE International Conference on Advances in Computing, Communication and Applied Informatics (ACCAI)*, pp. 1-7. <https://doi.org/10.1109/ACCAI53970.2022.9752577>
- [5] Jaydeep, P. (2021). Image acquisition and techniques. *International journal of advanced research in science. Communication and Technology*, 5(2): 225-229. <https://doi.org/10.48175/568>
- [6] Zhou, W., Ma, X., Zhang, Y. (2020). Research on image preprocessing algorithm and deep learning of iris recognition. *Journal of Physics: Conference Series*, 162(1): 1-9. <https://doi.org/10.1088/1742-6596/1621/1/012008>
- [7] Liu, H. (2022). Image segmentation techniques for intelligent monitoring of putonghua examinations. *Advances in Mathematical Physics*, 2022(6): 1-12. <https://doi.org/10.1155/2022/4302666>
- [8] Chowdhary, C., Acharjya, D. (2020). Segmentation and feature extraction in medical imaging: A Systematic Review. *Procedia Computer Science*, 167(1): 26-36. <https://doi.org/10.1016/j.procs.2020.03.179>
- [9] Jmour, N., Zayen, S., Abdelkrim, A. (2018). Convolutional neural networks for image classification. *International Conference on Advanced Systems and Electric Technologies (IC_ASET)*, Hammamet, pp. 397-402. <https://doi.org/10.1109/ASET.2018.8379889>
- [10] Cerutti, G., Tougne, L., Vacavant, A., Coquin, D. (2011). A parametric active polygon for leaf segmentation and shape estimation. In: *Bebis G et al. (eds) Advances in Visual Computing. ISVC 2011, Lecture Notes in Computer Science*, Springer, Berlin. Heidelberg.
- [11] Tian, Y., Lin, Z. (2012). Study on the methods of detecting cucumber downy mildew using Hyperspectral imaging technology. *Physics Procedia*, 33(1): 743-750. <https://doi.org/10.1016/j.phpro.2012.05.130>
- [12] Dornbusch, T., Andrieu, B. (2010). An image processing tool for an explicit description of lamina shape tested on winter wheat. *Computers and Electronics in Agriculture*, 70(1): 217-224. <https://doi.org/10.1016/j.compag.2009.10.009>
- [13] Larese, M., Craviotto, R.M., Arango, M.R., Gallo, C., Granitto, P.M. (2012). Legume identification by leaf vein images classification. *17th Iberoamerican Congress on Pattern Recognition. Buenos Aires, Argentina*, pp. 447-454. https://doi.org/10.1007/978-3-642-33275-3_55
- [14] Shantkumari, M., Uma, S.V. (2021). Grape leaf segmentation for disease identification through adaptive snake algorithm model. *Multimed Tools Appl.*, 80(1): 8861-8879. <https://doi.org/10.1007/s11042-020-09853-y>
- [15] Dey, N., Dutta, S., Dey, G., Chakraborty, S., Ray, R., Roy, P. (2014). Adaptive thresholding: A comparative study. *2014 International Conference on Control, Instrumentation, Communication and Computational Technologies*, pp. 1182-1186. <https://doi.org/10.1109/ICCICCT.2014.6993140>
- [16] Sharma, Ronak., Yadav, A. (2020). Contrast image enhancement using luminance component based on wavelet transform. *International Journal of Recent Trends in Engineering & Research*, 6(1): 22-29. <https://doi.org/10.23883/IJRTER.2020.6049.V2AAF>
- [17] Tao, li., Asari, V. (2004). An integrated neighborhood-dependent approach for nonlinear enhancement of color images. *International Conference on Information Technology: Coding Computing*, pp. 138-139. <https://doi.org/10.1109/ITCC.2004.1286612>
- [18] Li, Q., Wu, M. (2020). An improved hough transforms for circle detection using circular inscribed direct triangle. *International Congress on Image and Signal Processing, BioMedical Engineering and Informatics*, pp. 203-207. <https://doi.org/10.1109/CISP-BEI51763.2020.9263665>
- [19] Yuan, Y., Lo, Y. (2017). Improving dermoscopic image segmentation with enhanced convolutional-deconvolutional networks. *IEEE Journal of Biomedical and Health Informatics*, 23(2): 519-526. <https://doi.org/10.1109/JBHI.2017.2787487>
- [20] YunFei, Z., Zhang, X., Feng, W., Cao, T., Sun, M., Xiaobing, W. (2018). Detection of people with camouflage pattern via dense deconvolution network. *IEEE Signal Processing Letters*, 26(1): 29-33. <https://doi.org/10.1109/LSP.2018.2825959>
- [21] Zhou, H., Zhang, J., Han, A., Yang, H. (2018). Edge gradient feature and long distance dependency for image semantic segmentation. *IET Computer Vision*, 13(1): 53-60. <https://doi.org/10.1049/iet-cvi.2018.5035>