# Enhancing Flower Segmentation through Advanced Image Processing Techniques

***Abstract*— This paper proposes an efficient algorithm for segmenting flowers from complex backgrounds in images. The algorithm effectively addresses challenges such as pixel noise and foreground-background collisions by leveraging a pipeline of image processing techniques, including bilateral filtering, greyscale conversion, and saturation adjustment. Segmentation is achieved through K-means clustering and watershed algorithms, supplemented by morphological transformations for refinement. Empirical experimentation guides the selection of preprocessing techniques, leading to a robust pipeline for accurate flower segmentation. This work contributes to advancing automated flower segmentation for applications in botany, agriculture, and computer vision.**

**Keywords— Image Processing, Flower Segmentation, K-means Clustering, Watershed Algorithm, Morphological Operations**

## I. INTRODUCTION

Image processing remains a cornerstone in computer vision, offering extensive techniques for enhancing, analysing, and interpreting visual information. Discerning a flower from its background is a task of particular interest in botanical studies, agricultural management, and floriculture. The typical pipeline for segmenting a flower from its background begins with acquiring a high-quality image, followed by a series of preprocessing steps. These steps are designed to mitigate noise and normalise variations caused by lighting and shadow effects, thus preparing the image for more sophisticated operations. Following this, the image undergoes colour space transformations to amplify the contrast between the flower and its environment, leveraging the nuanced differences that are not perceptible in the traditional RGB space. Subsequently, segmentation algorithms come into play, often employing methods such as k-means clustering to partition the image into segments. The success of segmentation is typically quantified using median intersection over union (mIoU) metrics, providing a statistical measure of the precision of the segmentation. The outcome is a binary mask that highlights the flower, distinguishing it from a background that often consists of foliage, soil, and other environmental elements. This mask can then serve as a gateway to further analysis, such as feature extraction and classification, which may be used in automated identification systems or condition monitoring tools in intelligent farming solutions.

## II. LITERATURE REVIEW

Multiple image segmentation techniques could be applied in image processing. The most popular image thresholding method is Otsu, developed in 1979 as a simple and efficient algorithm for segmenting. For a simple algorithm, it provided sufficient segmentation of the images with low noise. When it comes to images with much noise, it is not easy to separate the foreground from the background [1]. The segmented image becomes ruined due to the algorithm segmenting parts of the image due to the algorithm detecting it as part of the noise. Research was conducted to find a way to automate the detection of cracks on the road. The research results achieved around 90% accuracy, with some modifications to the code [2].

The problem remains for images with high noise, where the segmentation becomes irregular. There are other methods of segmentation, such as K-means clustering, watershed and morphology, which are some of the widely used solutions for image segmentation.

Watershed uses edge detection by detecting the edges of the images by using a technique called “flooding”, which gets inspiration from geographical topography. It segments parts of the image into regions where it would then segment out images based on the gradient criteria that we may have set in the parameters. There are also multiple edge detection techniques, such as Canny, Sobel, or Prewitt.

K-means clustering is a segmentation technique developed by grouping similar datasets. Developed in 1955 for data analysis and measurements and later developed into an image segmentation technique that takes pixels with similar characteristics into clusters. With that, it could effectively segment out the foreground with the background [3].

Morphology is the process of removing imperfections after the segmentation process. It manipulates the pixels at the edge of the image. There are four methods in morphology: Dilation, Erosion, Opening, and Closing. Dilation is the process of expanding the size of the image. Erosion is the process of removing parts of the structuring elements of the image. Opening smoothens the edges around the image, removing noise and preserving the image. Closing combines different parts of the image with gaps or holes by filling them using similar characteristics of nearby pixels, maintaining the structure of the image [4].

## III. METHODOLOGY

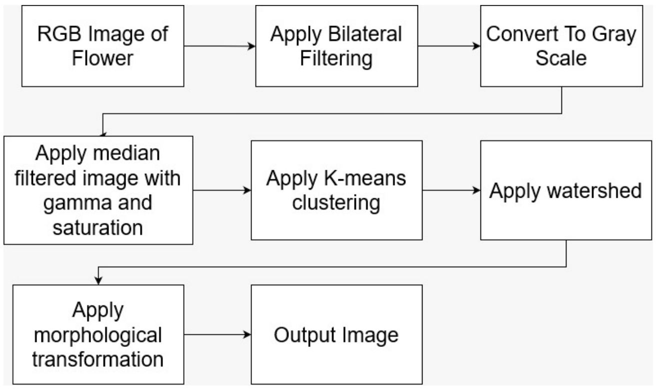
RGB Image of Flower
Apply Bilateral Filtering
Convert To Gray Scale
Apply median filtered image with gamma and saturation
Apply K-means clustering
Apply watershed
Apply morphological transformation
Output Image

Fig.1. The Complete Pipeline.

## A. *Initial Experimentation (Foundation of the Pipeline)*

The foundation of the pipeline was built through the analysis of many literature reviews. The research regarding disease detection in Maize Plants [5] and the ground truth segmentation paper [6] used a pipeline consisting of loading the image, running it through noise filtering, smoothing it to get it ready for segmentation, and then individual segmentation techniques. The first step is experimentation with edge detection algorithms.

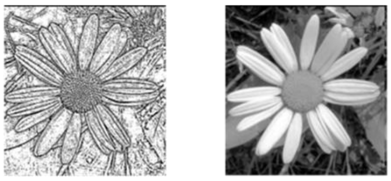
Fig. 2. Adaptive Thresholding versus Histogram Equalisation

As illustrated in Fig. 2, Adaptive Thresholding and Histogram Equalisation techniques reveal significant noise within the resultant images—the binary image outcome featuring considerable noise artefacts, posing challenges for subsequent noise removal processes.

The adaptive thresholding in Fig. 2. demonstrates that the processing of the images is more efficient since the threshold values are no longer constant and can be changed depending on the local characteristics of the image. However, it also comes with its own set of risks and challenges. One of the main risks is the introduction of excessive noise, which can significantly degrade the quality of the processed images. Moreover, the rigid nature of adaptive thresholding presents a challenge as input images must be similar to those used to be effectively modified using machine learning. Therefore, despite the potential benefits, the risks associated with adaptive thresholding may outweigh its advantages in many practical scenarios. As a result, a more reliable approach often involves focusing on improving image contrast and using thresholding methods that are more predictable and less susceptible to noise.

Canny is an edge detection algorithm that has multiple stages to the process, including a Gaussian Filter, a greyscale conversion, Gradient Calculation, Non-maximum suppression, Thresholding, and Edge Tracking [7]. Due to the thin edges, Canny's parameter tuning wasn't viable with a diverse set of images.

Sobel is a gradient based algorithm; it creates a mask over the image. It first creates a mask with vertical edges and then with the horizontal edges, combining them to create one mask for edge detection [8]. Sobel had similar results to Canny for the easy images, demonstrating an unmanageable amount of noise for the first step.

Prewitt is an edge detection based on the gradient computation, that has similar edge detection masking where it detects the horizontal and vertical edges. The difference being that like Sobel, it fails to identify diagonal edges, and higher values brought an unmanageable amount of noise [7].

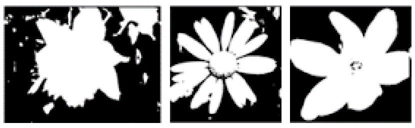
Fig. 3. Otsu's Segmentation Technique with Bilateral

Initially, Otsu's Method was chosen for segmentation based on its reputation in the literature. However, practical testing revealed drawbacks inherent to this approach. Otsu's method relies on the assumption of bimodal image histograms and is sensitive to noise, making it less effective for images with diverse lighting conditions. Additionally, varying noise levels necessitated different filtering strategies, complicating the segmentation process, especially for images with intricate details and varied colours. This complexity is evident in Fig. 3. where images with pronounced intricacy and colour diversity posed significant challenges for segmentation.

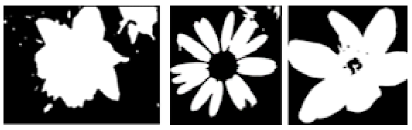
Fig. 4. K-Means Clustering Segmentation Technique with Bilateral Filtering.

The segmentation results obtained through K-means clustering showed promising outcomes, particularly in Fig. 4. In the first image, K-means successfully segmented the flower's edges and extracted more background components than Otsu's method. Similarly, in the second image, K-means neatly segmented the flower. However, in Fig. 3, the segmentation of the flower could have been more precise, with some background elements erroneously included. Refining the code could improve results compared to those obtained using Otsu's method.

K-means clustering is employed to cluster areas within the image, detecting pixels belonging to different clusters such as clusters 1, 2, or other designated values [10]. In the context of this study, the algorithm operates with a 2-cluster configuration, signifying the presence of two distinct areas (clusters) utilised in the segmentation process [10]. As outlined in the referenced paper, the image details are segregated into foreground (the flower) and background, as depicted in Fig. 4. Consequently, a 2-cluster setup is deemed the optimal choice for this algorithm [10].

The Watershed algorithm offers a beneficial approach to segmentation tasks. By leveraging watershed lines, this algorithm effectively delineates various regions within an image [11]. Specifically, Watershed uses the histogram terrain to define these lines, separating distinct areas. During the segmentation process, a threshold is applied to identify pivotal points, guiding the delineation of foreground and background regions to obtain a precise and accurate flower edge. Compared to traditional methods, the combination of thresholding and Watershed demonstrates increased efficiency in segmentation techniques [12].

Watershed detects image edges and accurately segments selected areas from the background [13]. This approach highlights image edges, resulting in smoother segmentation outcomes. By conceptualising the image as a topographical landscape, Watershed segments regions based on surrounding pixel values [13]. This methodology allows for precise segmentation by effectively delineating boundaries between different regions within the image.

Segmentation is pivotal in image processing, as highlighted in the referenced paper [14]. Integrating segmentation into the overall processing workflow enhances subsequent result accuracy and precision. Therefore, it is imperative to incorporate robust segmentation methodologies to bolster the efficacy of image processing endeavours.

### B. *The Image Processing Pipeline*

The first step was to modify the ground truth for an accurate comparison. Being in colour, it is impossible to get an accurate comparison with the binary output. The ground truth is converted into binary format, where the comparison flower is black, and the background is white. The ground truth colours are then inverted, as the output image has the flower in white and background in black. This process provides an accurate mIoU comparison with our binary output image.

Upon narrowing down a simple pipeline infrastructure with sample parameters from the Literature Review listed above, it became evident that fundamental techniques exist and are common in all pipelines:

- **Bilateral Filtering:** Bilateral filtering stands out as a superior alternative to Gaussian filtering, primarily due to its ability to furnish a smoother image while preserving edges more effectively. Unlike the Gaussian filter, which applies uniform filter values across the entire image, bilateral filtering modulates its filter values based on spatial distance and intensity differences [15]. This nuanced approach prevents excessive blurring, thus reducing the likelihood of pixel loss. This method ensures heightened image fidelity and clarity by employing a sigma value of 75 for both colour and spatial domains.

- **Greyscale Conversion:** The conversion of images from the BGR colour space to greyscale is an indispensable preprocessing step. Greyscale conversion enhances the differentiation between foreground and background elements by eliminating colour information and retaining only luminance. Notably, lighter pixels are mapped to white values, while darker pixels are represented as black. Nonetheless, pixel values may still overlap in specific scenarios, posing challenges for subsequent segmentation tasks.

- **Median filtering:** Median filtering plays a crucial role in refining segmented images due to its efficiency in noise suppression. Notably, empirical findings from research literature advocate for the utilisation of a kernel size (k-value) of 7 [16]. Experimentation and optimisation determined that a k-value of 5 yields optimal outcomes, manifesting a notable enhancement ranging from 1% to 2% in segmented image quality. This finer-tuned approach reduces noise without compromising image fidelity or edge preservation.

An experimental exploration was undertaken in response to this issue, focusing on Gaussian Filtering, Median Filtering, and Bilateral Filtering as potential noise mitigation strategies. The experimentation revealed that a sequential application of Gaussian and Bilateral filtering yielded the most favourable mean Intersection over Union (mIoU) result, demonstrating a notable 2% enhancement in image comparison metrics.

$$BF[I]_p = \frac{1}{W_p} \sum_{q \epsilon S} G\sigma_s(||p-q||) G\sigma_r(|I_p - I_q|) I_q \quad (1)$$

$BF[I]$ represents the product of the image's pixel output and intensity. $W_p$ is the normalisation factor, which helps the filtering scaling for all pixels. $\sum_{q\epsilon S}$ is the summation of the pixel value within the kernel window. $G\sigma_s$ represents the size of the kernel window. $p, q$, represents the $X, Y$ coordinates on a plane. $G\sigma_r$ is the range which takes the kernel's size and determines the filtering's intensity. $I_p$ represent the target pixel value. $I_q$ are the neighbouring pixels.

Eq. (1). demonstrates the application of bilateral filtering, showcasing the modified Gaussian smoothing formula. This technique incorporates parameters such as the diameter of each pixel neighbourhood, the sum filter of the colour space, and the sum of filters of the coordinate space.

The utilisation of bilateral filtering effectively mitigates unwanted noise within floral images, enhancing their visual quality and facilitating subsequent analytical tasks. Higher values result in smoother images by incorporating a more comprehensive range of pixel colour mixtures, which might lead to a more blended output. Conversely, it affects pixel coordination in the spatial domain, causing blurring effects that help preserve image edges. Eq. (1). assists in filtering out more of the noise with a higher intensity but preserves the edges. Modifying the parameters adjusts the algorithm's intensity in reducing the image's noise while keeping it in the edges of the foreground. It is slower to process the images due to it having to modify each pixel value to a weighted average, resulting in better filtering of noises.

TABLE 1: RESULTS ON MEDIAN AND GAUSSIAN FILTERS.

| **Image Name** | **mIoU comparison percentage of similarity between the output and ground truth** | | |
|---|---|---|---|
| | ***Median*** | ***Gaussian*** | ***Mean*** |
| Easy 1 | 81.5% | 81.4% | 81.4% |
| Easy 2 | 83.4% | 83.3% | 83.3% |
| Easy 3 | 80.8% | 80.7% | 80.5% |
| Medium 1 | 74.7% | 73.7% | 73.7% |
| Hard 3 | 90% | 89% | 89% |

$$F_{((x,y))} = median(P_k) \ (2)$$

$F_{((x,y))}$ is the result of the median value, which is applied to the centre of the neighbourhood. $median\ (P_k)$ represents the median of the pixel value within the kernel window.

Using odd values for 'k' is crucial to ensure a central pixel in the neighbourhood. A more enormous 'k' value encompasses more pixels in the neighbourhood, potentially resulting in increased image smoothing. Eq. (2). is used to filter out the small noises initially. The k-value correlates with the window size in which the kernel is processed by selecting a higher k-value. It would allow more pixels for computation and takes the sum of all the pixels in the window and the median of that summation. Setting the kernel window would determine the flower edges' smoothness and the pixel image's preservation. Once the kernel window has been set, it will go through the Median Filtering Formula.

Setting 'k' to 5 through experimentation in Eq. (2). resulted in optimal filtering performance. This choice strikes a balance between capturing sufficient neighbouring pixels for effective smoothing while preserving the essential features of the image. Adjusting the 'k' value appropriately allows tailoring the filtering process to the specific characteristics of the input image.

The introduction of gamma and saturation adjustments as supporting values for the median filter has proven to be a valuable enhancement in our image processing pipeline. Gamma correction effectively brightens general pixels, while saturation adjustment balances colour distribution. Through experimentation with a range of values spanning from 0.1 to 1.2, the gamma values and saturation have been set to 0.95 and 0.94 to yield optimal results. Specifically, these adjustments led to a noticeable improvement of 1% to 2% in output quality, particularly evident in easy-1 and medium-1 scenarios. Notably, values at the extremes of the range (e.g., below 0.6 or above 0.9) yielded undesirable outcomes, either failing to produce significant enhancements or causing over-brightness that merged background elements with the foreground. Our chosen values strike a delicate balance, effectively brightening the flowers while preserving segmentation integrity. Further increases in these values could lead to image contraction and diminished results. Moreover, incorporating median filtering as a refinement step following bilateral filtering further enhances the overall quality of our image processing pipeline.

### C. *ROI Misimplementation*

During the implementation of Region of Interest (ROI) and Contours on the image, the Hue-Saturation-Value (HSV) parameters were adjusted to detect white and yellow images while disregarding other colours. The approach involved detecting the image colour and tuning custom HSV values accordingly. Initially, the implementation adjusted for the colour yellow but soon encountered challenges due to the diverse spectrum and range of the images. It became apparent that custom HSV values must be tailored for each colour combination, leading to a hard-coded implementation. Fig. 5. illustrates the limitations associated with such an approach.

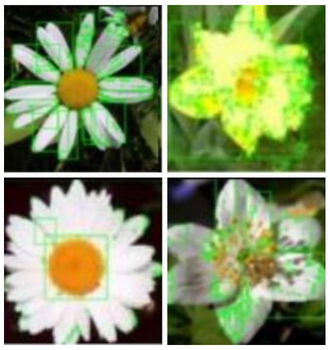

Fig. 5. Region of Interest for White and Yellow Flowers

To address this issue, the HSV partitioning system creates an identification mechanism for white and yellow flowers, which uses a mask-based filtering approach. However, this approach led to the system being hard coded for a specific range of yellow and white flowers, rendering it impractical. Subsequently, to modify the preprocessing, the images were converted to greyscale. This adjustment simplifies the filtering and segmentation processes, facilitating a more balanced flower segmentation approach. Through testing, these results were worse, and thus subsequently discarded.

### D. *Segmentation*

K-means clustering is a primary segmentation algorithm that considers the region of interest by the cluster region. It creates clusters based on similar characteristics by calculating the pixels, finding the mean, and maintaining the clusters. Pixels that are farther away from the mean are discarded [17].

The effectiveness of K-means clustering in image segmentation largely depends on the selection of the number of clusters (k-value), along with other algorithmic parameters such as epsilon and the maximum number of iterations. The k-value is pivotal as it directly influences the granularity of the segmentation—determining how many distinct regions are identified within the image.

- **Number of Clusters (k-value):** Selecting an optimal k-value is crucial for achieving the desired segmentation accuracy. Empirical analysis has demonstrated that a k-value of 2 often produces the most effective segmentation results for binary classification tasks, such as distinguishing flowers from their background. This choice is based on the typical distribution of pixel intensities where two significant groups (foreground and background) are predominant.
- **Termination Criteria:** Two critical termination criteria used in K-means are epsilon and the maximum number of iterations. Termination Criteria Epsilon, a threshold for the change in centroid position, determines the algorithm's sensitivity to changes between iterations. Setting epsilon to 0.2 ensures that the algorithm stops when centroids move less than this threshold,

indicating stabilisation. Limiting the number of iterations to 10 helps prevent excessive computational load while ensuring sufficient convergence of the centroids to their optimal positions.

Particular challenges were observed while implementing K-means clustering for image segmentation, largely due to its algorithmic reliance on colour intensity to define clusters. This often led to segmentation errors under specific conditions:

- **Inconsistent Results in Complex Backgrounds:** For images with highly textured or similarly coloured backgrounds, K-means clustering occasionally misclassified background pixels as part of the foreground. This issue was particularly prevalent in the 'medium' and 'hard' difficulty categories where background complexity increased. Fig. 4. demonstrates the issue exceptionally well, where the noise can lead to misclassification.
- **Sensitivity to Colour Variations:** Images with subtle colour variations within or between the flower and the background sometimes resulted in incomplete segmentation. This was due to K-means relying on colour intensity clustering, which can fail to distinguish between closely matched colours.

A clustering approach was adopted for image segmentation, utilising a fixed value of k = 2. The determination of this parameter value was derived from a systematic experimentation process. Specifically, evaluations were conducted across various values of k, including 3, 4, 5, and 7. However, empirical findings indicated that these alternative values yielded worse outcomes when the k-values were higher. Notably, employing k = 2 resulted in a segmentation that removes part of the flower, blending it into the background and diminishing the overall segmentation quality. Subsequently, the ensuing images exhibited undesired characteristics, contributing to a decreased overall performance score.

The Watershed algorithm was applied following K-means clustering to refine the segmentation boundaries. Based on the morphological concept of 'flooding', this technique proved effective in delineating the precise edges of flowers, mainly when the initial segmentation from K-means was broadly accurate.

- **Enhanced Edge Detection:** Watershed improved the edge delineation, especially in images where the initial K-means output clearly distinguished foreground and background. It effectively resolved overlaps and isolated the flower edges from complex backgrounds.
- **Dependency on Initial Mask Quality:** The effectiveness of the Watershed algorithm significantly depended on the quality of the binary mask provided by K-means. In cases where K-means failed to identify the foreground accurately, Watershed could not compensate, resulting in poor overall segmentation.

### E. *Morphology*

Applying morphology using erosion and dilation improves mIoU's image. The erosion will erase the image's border, which is small and smaller than the minimum values of the neighbourhood area. Therefore, the remaining parts of the image processed through the previous filter and segmentation will be significantly erased. Besides that, dilation has the opposite effect on the image, enriching the border of the image and making it more visible and thicker [17]. This is the final step to complete the segmentation part of the process.

Using watersheds proved to improve the areas surrounding the flower. By combining erosion and dilation, the watershed process will be more accurate and efficient when running through the segmented part, refining the edges of the flower to make a cleaner cut.

## IV. Result and Discussion

After preprocessing the flower image, the image of the flower is converted into a binary version. It is then compared with a binary version of the ground truth. The ground truth is binary, and then the colour is inverted.

As a result, using the watershed with k-means clustering resulted in a similar accuracy for most of the flowers except for medium_1.

TABLE I. RESULTS OF THE FINAL PROCESSING RESULTS.

| Final Processing Results | *MIOU Results* |
|---|---|
| Easy 1 | 81.5% |
| Easy 2 | 83.4% |
| Easy 3 | 80.8% |
| Medium 1 | 74.7% |
| Medium 2 | 84.4% |
| Medium 3 | 83% |
| Hard 1 | 88.9% |
| Hard 2 | 87.8% |
| Hard 3 | 90% |

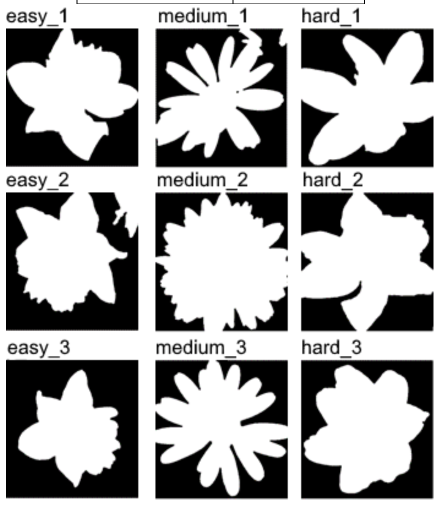

Fig. 6. The Final Output

As illustrated in Fig. 6. the segmentation pipeline encountered its most significant challenge with the medium_1 flower image, particularly in distinguishing the foreground flower from the background flower. This difficulty stemmed from the algorithm inadvertently excluding one of the flowers in the background, contradicting the ground truth where both flowers are included. The bottom left of medium_1 demonstrates the flower being bloated due to excessive dilation. However, the algorithm demonstrated more consistent performance across other flower images, with particularly notable results in the easy and hard categories. The success in these categories can be attributed to the relatively cleaner background, which reduced noise and facilitated more precise segmentation of the foreground from the background. Conversely, in the medium category, significant noise in the background posed challenges for accurate segmentation, leading to difficulties distinguishing foreground from background. These observations underscore the importance of considering image complexity and noise levels when evaluating segmentation algorithms, highlighting areas for potential improvement in handling diverse image characteristics.

## Conclusion

In conclusion, our research underscores the effectiveness of integrating watershed and k-means clustering techniques for flower image segmentation, outperforming the traditional Otsu thresholding method. By leveraging the complementary strengths of both approaches, finer segmentation of flower margins against the background is achieved. This enhanced capability in edge detection facilitates more accurate segmentation of foreground flowers from the background, contributing to improved overall segmentation quality. The findings highlight the significance of utilising advanced clustering techniques in image processing tasks, particularly in scenarios where precise segmentation is crucial. Further exploration and refinement of hybrid clustering methods hold promise for advancing the state-of-the-art in flower image analysis and related fields.

## References

[1] C. Sha, J. Hou, and H. Cui, ‘A robust 2D Otsu’s thresholding method in image segmentation’, Journal of Visual Communication and Image Representation, vol. 41, pp. 339–351, 2016.

[2] A. Akagic, E. Buza, S. Omanovic, and A. Karabegovic, ‘Pavement crack detection using Otsu thresholding for image segmentation’, in 2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), 2018, pp. 1092–1097.

[3] A. K. Jain, ‘Data clustering: 50 years beyond K-means’, Pattern Recognition Letters, vol. 31, no. 8, pp. 651–666, 2010.

[4] Goyal, M. (2011). Morphological image processing. IJCST, 2(4), 59.

[5] M. A. Mohd Yusof and A. Nazari, “The Disease Detection for Maize-Plant using K-Means Clustering”, *EEEE*, vol. 2, no. 2, pp. 834–841, Nov. 2021, Accessed: Apr. 25, 2024. [Online]. Available: https://publisher.uthm.edu.my/periodicals/index.php/eeee/article/view/4412

[6] M. Henke, K. Neumann, T. Altmann, and E. Gladilin, ‘Semi-Automated Ground Truth Segmentation and Phenotyping of Plant Structures Using k-Means Clustering of Eigen-Colors (kmSeg)’, *Agriculture*, vol. 11, no. 11, 2021.

[7] Samina, “What is canny edge detection?,” Educative, https://www.educative.io/answers/what-is-canny-edge-detection (accessed Apr. 24, 2024).

[8] Tutorialspoint, “Sobel operator,” Tutorialspoint, https://www.tutorialspoint.com/dip/sobel_operator.htm (accessed Apr. 24, 2024).

[9] GeeksforGeeks, “Edge detection using Prewitt, Scharr and Sobel operator,” GeeksforGeeks, https://www.geeksforgeeks.org/edge-detection-using-prewitt-scharr-and-sobel-operator/ (accessed Apr. 24, 2024).

[10] H. Tariq and s. M. A. Burney, ‘K-Means Cluster Analysis for Image Segmentation’, International Journal of Computer Applications, vol. 96, 06 2014.

[11] A. Seal, Das, P. Sen, and International Journal Of Computer Science and Information Technologies, “Watershed: An Image Segmentation Approach,” Watershed: An Image Segmentation Approach, vol. Vol 6, no. 3, Art. no. ISN:0975-9646, May 2015, [Online].

[12] N. Venu and A. Kumar, ‘Comparison of Traditional Method with Watershed Threshold Segmentation Technique’, 07 2022.

[13] Y. Wu and Q. Li, “The algorithm of watershed color image segmentation based on morphological gradient,” Sensors, vol. 22, no. 21, p. 8202, Oct. 2022, doi: 10.3390/s22218202.

[14] W. N. Jasim and R. H. Mohammed, “A survey on segmentation techniques for image Processing,” Iraqi Journal for Electrical and Electronic Engineering/Al-Mağallaẗ Al-ʻirāqiyyaẗ Al-handasaẗ Al-kahrabāʼiyyaẗ Wa-al-ilikttrūniyyaẗ, vol. 17, no. 2, pp. 73–93, Aug. 2021, doi: 10.37917/ijeee.17.2.10.

[15] C. Tomasi and R. Manduchi, ‘Bilateral filtering for gray and color images’, in Sixth International Conference on Computer Vision (IEEE Cat. No.98CH36271), 1998, pp. 839–846.

[16] I. Patel, S. Patel, and A. Patel, “Analysis of various image preprocessing techniques for denoising of flower images,” International Journal of Computer Sciences and Engineering, vol. 6, no. 5, pp. 1111–1117, May 2018, doi: 10.26438/ijcse/v6i5.11111117.

[17] D. Chudasama, T. Patel, S. Joshi, and G. I. Prajapati, “Image Segmentation using Morphological Operations,” International Journal of Computer Applications, vol. 117, no. 18, pp. 16–19, May 2015, doi: 10.5120/20654-319