

Efficient Cluster Number Selection in K-Means using CIELAB Color Space for Image Segmentation

Xuanzhi Zhang, *Member, IEEE*

Abstract— In the field of image analysis, computer algorithms must recognize and extract essential features or patterns within visual data to accurately represent objects, scenes, or activities. Color information is a critical aspect of object classification, as color differences enable humans to identify the presence of one or more objects. This research introduces a novel method for determining the optimal number of color clusters in CIELAB color space images, using mathematical expressions and K-Means as image segmentation method. Our approach combines perceptual lightness, red, green, blue, and yellow histogram standard deviations, along with the number of local maximum points, by employing a non-linear equation to calculate the ideal number of clusters. Experimental results reveal that the proposed method surpasses widely used techniques such as the Elbow Method and Gap Statistic Method in terms of time. Moreover, our algorithm demonstrates exceptional accuracy in processing visual data within real-world environments, maintaining computational efficiency even as the number of clusters increases.

Keywords—CIELAB; Clustering; Color Histogram; K-Means; Image Segmentation; Non-linear Transformation

I. INTRODUCTION

Computer vision aims to enable computers to interpret and understand visual information from the world, much like humans use their eyes and brains to perceive and make sense of their surroundings. In object recognition, color information serves as a crucial feature for identifying distinct regions within an image. Consequently, enhancing object recognition accuracy necessitates a time-efficient method that can analyze inherent properties of RGB information in visual data in real-time, without relying on large datasets, and segment images based solely on color information.

K-means[1] is a widely employed unsupervised machine learning algorithm designed to categorize data into distinct groups, predicated on similarities among their attributes. Within the K-means framework, the user determines the quantity of clusters (k), and the algorithm allocates each data point to one of these k clusters by evaluating the similarity of its attributes to the respective cluster centroids.

This paper introduces a mathematical expression that addresses the limitations of existing automatic k -value calculation algorithms for determining the optimal number of

clusters in clustering techniques. The proposed method overcomes challenges associated with traditional techniques such as the Elbow Method[2] and Gap Statistic[3], which are sensitive to image size, high computational complexity, and the need for a predefined maximum number of k . Furthermore, this paper presents a comprehensive overview of the advantages of the proposed algorithm and demonstrates its effectiveness in enhancing the accuracy and efficiency of clustering techniques for computer vision applications.

II. CIELAB COLOR SPACE

In our method, color serves as the sole feature; Therefore, the ability to extract more information from color is crucial. Increasing the number of dimensions to describe each color can enhance color discrimination. As depicted in Figure 1, the conventional RGB color space exhibits cubic geometry. However, a more irregular geometry is anticipated to improve the distinction between colors.

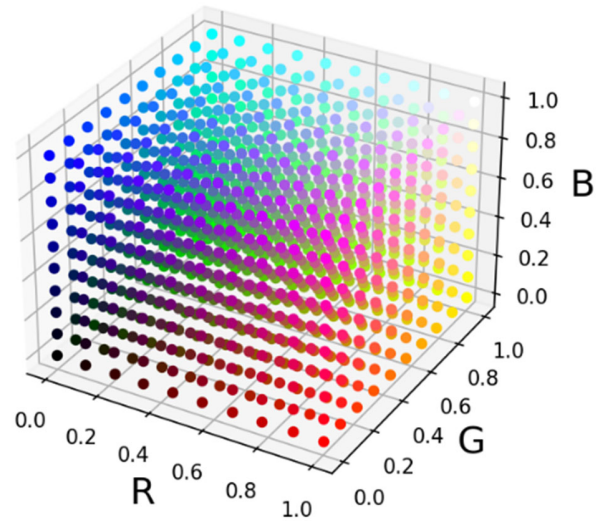


Fig. 1. 3-Dimension RGB color space

CIELAB[4] is a perceptually uniform color space, which means that the distances between colors in the color space are proportional to the differences perceived by the human visual system. As illustrated in Figure 2, the distribution of colors in the CIELAB space takes the form of a distorted cylinder. The L^* value represents the lightness of the color, ranging from 0 (black) to 100 (white), approximating the perceived brightness of colors. The a^* value corresponds to the green-red axis, indicating the position of the color on this axis. A negative value of a^* signifies a greenish tint, while a positive value of

Xuanzhi Zhang is with the Department of Physics, the Ohio State University, Columbus, OH 43210 USA (e-mail: zhang.10419@osu.edu).

Color versions of one or more of the figures in this article are available online at <http://ieeexplore.ieee.org>

0009-0001-0854-8746

a^* represents a reddish tint. The b^* value corresponds to the blue-yellow axis, with negative values of b^* implying a blue tint and positive values signifying a yellow tint.

The distinctive geometry and perceptual uniformity make CIELAB an ideal color space for calculating the number of color segments and accurately simulating human color perception.

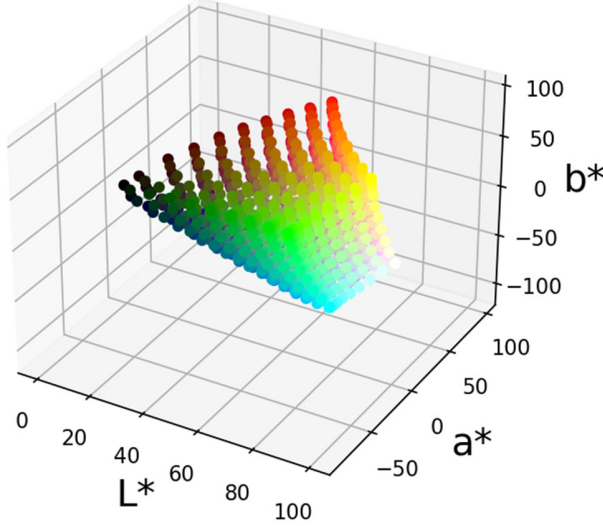


Fig. 2. 3-Dimension CIELAB color space

III. MATH

A. Proposed Method

The method proposed in this paper can be divided into three parts. The first part is to transform three-channel RGB color into three-channel LAB and extract the histogram of L, A and B respectively. The second part is to calculate the optimal standard deviation and number of local maxima from three channels respectively as parameters. Finally, the best number of clusters is determined using the CIELAB Color Segment Estimation (CCSE) function, as shown below:

$$k = \left\lceil \ln \left(\frac{\sigma_L}{n_L} \right) \right\rceil \max(n_a, n_b) + \min(n_a, n_b) \quad (1)$$

In this equation, σ_L denotes the standard deviation of L histogram and n_L , n_a , n_b represent the number of local maxima extracted from the L, a, and b histograms in the second part. $\max(n_a, n_b)$ refers to the larger value between n_a and n_b , while $\min(n_a, n_b)$ indicates the smaller value.

The detailed steps of the designed method can be found in the pipeline presented in Figure 3. It is important to note that this method specifically requires input images in the three-channel RGB color format.

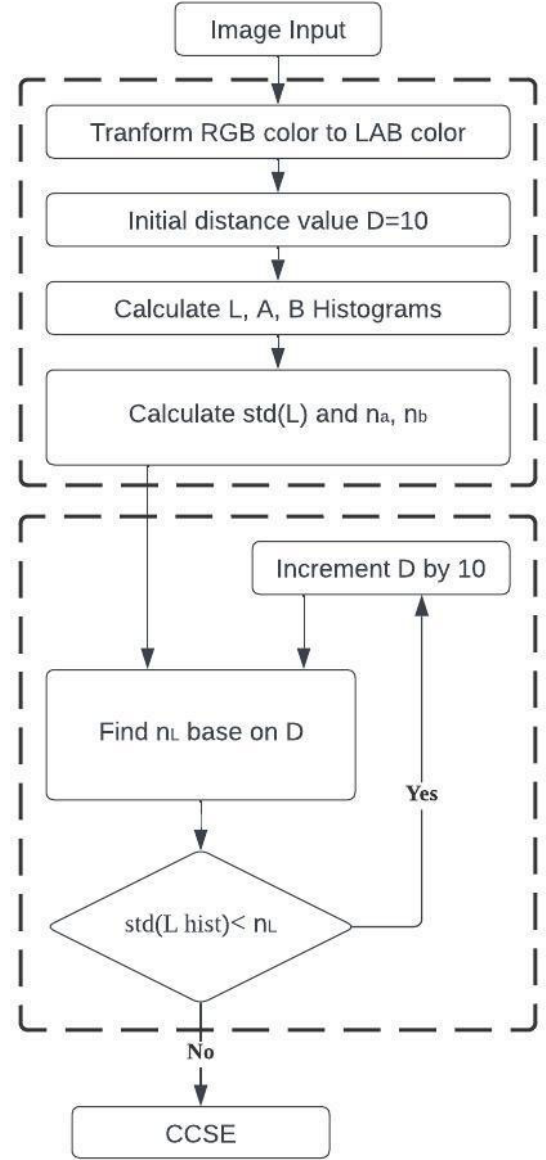


Fig. 3. Designed method pipeline

B. Color Histogram

Color is usually represented by color histogram, color correlogram, color coherence vector, and color moment under a certain color space. The color histogram serves as an effective representation of the color content of an image if the color pattern is unique compared with the rest of the data set.[5].

We extract information from the L, a, and b histograms by examining the standard deviation and the number of local maxima[6][7]. These metrics provide insights into the following aspects and explain the idea behind CCSE function:

1. **Lightness distribution:** The L histogram's standard deviation reveals the range of lightness values in the image. A higher standard deviation signifies a broader range, while a lower value indicates a narrower range.

2. **Color distribution:** Standard deviations in the a and b histograms reflect the range of colors present in the image. Higher values suggest a more diverse color palette, while lower values indicate a more limited one.
3. **Peak separability:** The number of local maxima in the L, a, and b histograms informs the separability of different lightness and color levels, which is useful for image segmentation or object recognition. More local maxima denote distinct peaks, while fewer maxima suggest closely clustered peaks.
4. **Image complexity:** A combination of the standard deviation and the number of local maxima in the L, a, and b histograms provides an indication of overall image complexity in terms of lightness and color distribution. Images with higher standard deviations and more local maxima are considered more complex, while those with lower values are simpler.

In conclusion, color histogram analysis allows for a comprehensive understanding of an image's lightness and color distribution, peak separability, and overall complexity. This information can be invaluable for various image processing tasks, such as segmentation and object recognition.

C. CCSE

The term $\left[\ln \left(\frac{\sigma_L}{n_L} \right) \right]$ calculates the logarithm of the ratio between the standard deviation of the L histogram and the number of local maxima in the L histogram. This ratio reflects the relative spread of lightness values and the degree of separability between different lightness levels in the image. Taking the logarithm of this ratio helps normalize the scale of the values, ensuring that the final estimation is less sensitive to extreme values. logarithm is rounded down to make sure that the number of clusters is an integer. To keep normalized cluster number positive and avoid too small values of σ_L because the colors in the picture all belong to the same hues, we use the algorithm outlined by the lower dashed boarder in Figure 3 to solve this problem.

By incorporating $\max(n_a, n_b)$ into product term, the CCSE function accounts for the channel with the most distinct local maxima, which corresponds to the channel with the most dominant color variation. In contrast, using $\min(n_a, n_b)$ would emphasize the channel with the least color variation, which might lead to an underestimation of the true number of color segments required for effective image segmentation.

Adding $\min(n_a, n_b)$ to the expression allowing the contribution of the less dominant color channel to the overall color distribution is not entirely disregarded. While it is essential to prioritize the channel with the most significant color variation, considering the less dominant channel helps achieve a more accurate estimation of the required number of color segments.

D. Algorithm

The Algorithm 1 shown below is a detailed coding for the pipeline shown in Figure 3.

Algorithm 1 CIELAB Color Segment Estimation

Input: 2D array *image* containing the color pixel values

Output: number of cluster *k*

```

1. function CCSE(image):
2.   lab_image = RGB2LAB(image)
3.   L, a, b = split(lab_image)
4.   HistL ← histogram(L, bins=256)
5.   Hista ← histogram(a, bins=256)
6.   Histb ← histogram(b, bins=256)
7.   D ← 10
8.   MaxL ← Find_Local_Maxima (HistL, D)
9.   Maxa ← Find_Local_Maxima (Hista, D)
10.  Maxb ← Find_Local_Maxima (Histb, D)
11.  while std(HistL) < len(MaxL) do
12.    D ← D+10
13.    MaxL ← Find_Local_Maxima (HistL, D)
14.  std_dev ← std(HistL) / len(MaxL)
15.  log_std_dev ← log(std_dev)
16.  log_max_peaks ← MAX(len(Maxa), len(Maxb))
17.  k ← floor(log_std_dev * log_max_peaks) +
      MIN(len(Maxa), len(Maxb))
18.  return k
19. end function

```

E. Time Complexity

The time complexity of the given function can be analyzed by looking at the time complexity of each of its operations. Given that the histogram size is no more than 256 bins, the time complexity of finding peaks using *Find_Local_Maxima* can be treated as constant. Therefore, the time complexity of the function can be simplified as follows: assigning image variable has a time complexity of $O(1)$, converting the image from RGB to LAB color space has a time complexity of $O(N)$, where N is the total number of pixels in the image, calculating histograms has a time complexity of $O(N)$, finding peaks has a time complexity of $O(1)$, the while loop in the function has a worst-case time complexity of $O(N)$, and the mathematical operation to determine the number of clusters has a time complexity of $O(1)$. Therefore, the overall time complexity of the function is $O(N)$, where N is the total number of pixels in the image.

IV. SEGMENTATION

A. Visual Validation

The performance of image segmentation using the CCSE algorithm can be validated by visually comparing the segmented images produced using the optimal number of clusters (k) calculated by the CCSE method against those produced using existing methods, such as the Elbow method and Gap Statistic method. Since both the Elbow method and Gap Statistic method require a range of k values to test, we can set an initial maximum value K_{\max} as the range of k to test. for all test cases. In this instance, we choose a reasonable value of $K_{\max}=20$ for all testing scenarios. This enables us to explore the effectiveness of the CCSE algorithm and compare its performance with well-

0009-0001-0854-8746

established cluster estimation methods.

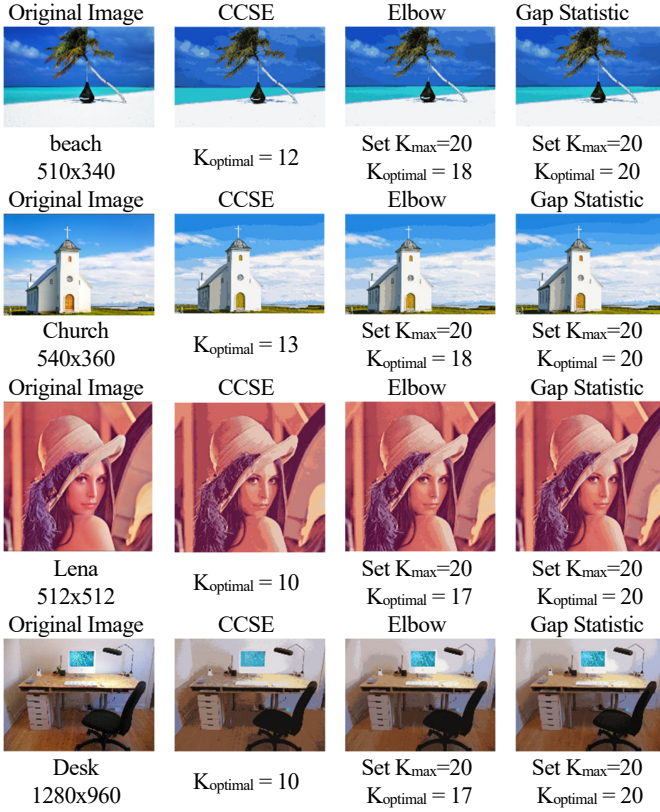


Fig. 4. Visual comparison of segmented images using k value calculated by CCSE, Elbow method and Gap Statistic Method.

B. Quantitative Validation

A further validation can be done by Color Reconstruction Error-based Segmentation Evaluation (CRESE) which evaluates the performance of a color-based image segmentation method, such as the CCSE algorithm, by measuring the color reconstruction error in the segmented image. It calculates the color differences between the true color of each pixel and the average color of the corresponding cluster (label) in the segmented image and sums up the color differences for all pixels. A lower color reconstruction error indicates better performance of the segmentation method in estimating the number of color-based clusters.

Let O be the original image of size $H \times W \times 3$ (height, width, and RGB color channels), and let S be the segmented image of the same size with averaged colors per cluster.

Let $C = \{c_1, c_2, \dots, c_n\}$ be the set of unique colors in S , where n is the number of unique colors. For each color c_i in C , define M_i as the binary mask of size $H \times W$ that has 1s where S has color c_i and 0s elsewhere.

The color reconstruction error E is given by:

$$E = \sum_{i=1}^n \sum_{h=1}^H \sum_{w=1}^W M_i(h, w) * \|O(h, w) - c_i\|_2 \quad (2)$$

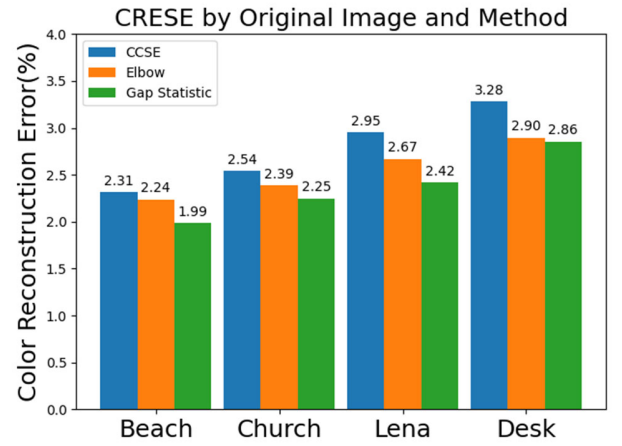
Where $\|\cdot\|_2$ denotes the Euclidean distance. And the normalized error E to a percentage scale can be calculated as

following:

$$E_{normalize} = \frac{E * 100}{H * W * \sqrt{255^2 * 3}} \quad (3)$$

During the experiment, we found that enhancing image contrast also leads to an increase in the number of k . Since the contrast enhancement process has an $O(1)$ time complexity, the calculation time of CRESE using the original image will not be affected. Consequently, based on the CRESE, we can obtain the validation plot in terms of normalized error vs. the number of clusters (k) for both enhanced images and original images, as shown in Figure 5.

(a)



(b)

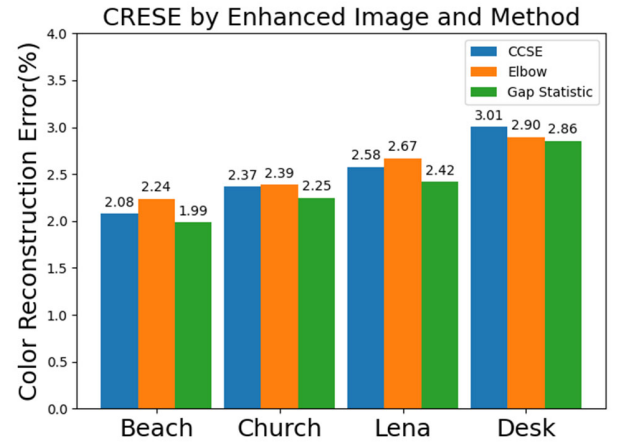


Fig. 5. (a) color reconstruction error when processing the original images. (b) color reconstruction error when processing the original images with enhanced contrast.

C. Computational Efficiency

In the previous section, we evaluated the accuracy of the three algorithms (CCSE, Elbow Method, and Gap Statistic Method) when processing images of different resolutions. In this section, we focus on comparing their computation times in the context of image segmentation. To do this, we use a high-resolution beach image without enhancement, having a size of 3840*2160 pixels as the test case. This unmodified image allows us to investigate the impact of image size parameters on the performance of each

0009-0001-0854-8746

algorithm in their default configurations. To do this, we create resized versions of the original image, maintaining its aspect ratio while reducing its dimensions to 10%, 25%, 33%, 50%, and 75% of the original size. This results in a diverse set of test images with different pixel counts, allowing us to thoroughly examine the efficiency of each algorithm in handling a range of image sizes.

To maintain a fair comparison, we set the maximum number of possible clusters (k) to 20 for both the Elbow Method and Gap Statistic Method. For each resized image, we measure the time taken by each algorithm to estimate the optimal number of clusters and analyze how the computational complexity varies with image size. By comparing the computation times, we can evaluate the efficiency of each algorithm in processing images of different dimensions and identify the most suitable method for real-world applications, where image size may vary significantly. Furthermore, this comparison will highlight the scalability and adaptability of the CCSE in handling diverse image sizes while maintaining accurate and effective segmentation results.

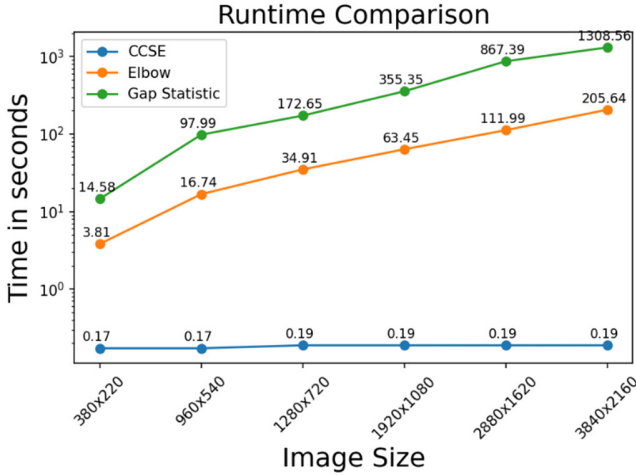


Fig. 6. Comparison of runtime performance for the CCSE, Elbow Method, and Gap Statistic Method when processing images with different resolutions.

D. Sensitivity Analysis

Sensitivity analysis is a crucial aspect of evaluating the performance of clustering algorithms, as it helps determine the robustness of each method against changes in their input parameters. In the context of our image segmentation study, we can assess the sensitivity of the Elbow Method and Gap Statistic Method by varying the maximum number of clusters (k) and observing how the clustering results and the calculated metrics are affected. Similarly, for the CCSE algorithm, we can examine how varying the distance parameter (D) influences the estimated number of clusters and the overall segmentation quality. By systematically altering these parameters and analyzing the resulting changes in the clustering quality, we can gain insights into the stability and adaptability of each method. A more robust algorithm will exhibit less sensitivity to parameter changes and produce more consistent results across a wider range of parameter values. In our study case, we always set parameter D for CCSE equal to K_{max} for Elbow and Gap Statistic Methods, starting from 5 to 40 with an increase step of 5.

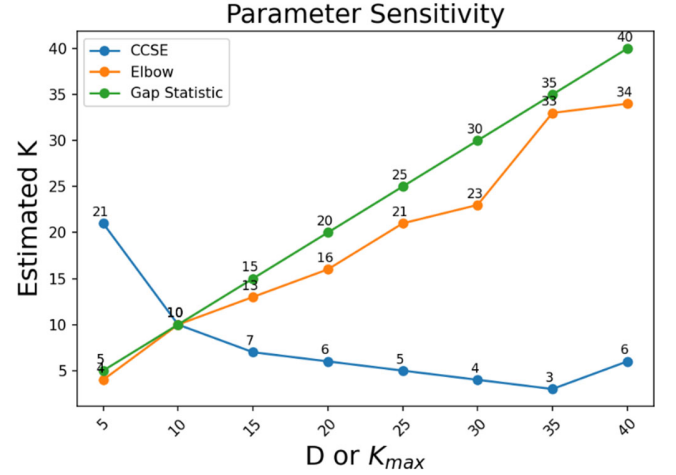


Fig. 7. Sensitivity to variations in input parameters (D for CCSE, K_{max} for Elbow and Gap Statistic methods)

Gap Statistic method shows a linear relationship between K_{max} and the estimated k value, suggesting a high sensitivity to changes in K_{max} . The Elbow Method demonstrates a non-linear relationship and somewhat less sensitivity to changes in K_{max} compared to the Gap Statistic method. However, the changes in k are still quite significant as K_{max} increases.

In contrast, the CCSE method exhibits a more significant decrease in k as the distance parameter D increases. This suggests that the CCSE method is less sensitive to parameter changes compared to the other two methods, as it shows more consistent results across a wider range of parameter values.

Based on this analysis, the CCSE algorithm appears to have the best robustness among the three methods, as it demonstrates the least sensitivity to changes in its input parameters.

IV. CONCLUSION

After analyzing results demonstrate above, we can say that the proposed model and CCSE algorithm demonstrates superior performance in various aspects of image segmentation, including runtime efficiency, robustness, visual validation, and quantitative validation.

First, the segmented images produced by the CCSE algorithm appear visually like those produced by the other methods, demonstrating that the algorithm effectively captures the essential characteristics of the input images and provides satisfactory clustering results. We further use the CRESE metric on both original and contrast-enhanced images, the CCSE algorithm shows competitive performance compared to the Elbow method and Gap Statistic method. For original images, CCSE errors range from 2.31 to 3.28, while for enhanced images, the errors range from 2.08 to 3.01. These results indicate that the CCSE algorithm can provide accurate segmentation results, even when dealing with images with different levels of contrast.

On the other hand, the runtime comparison shows the CCSE algorithm significantly outperforms both the Elbow method and Gap Statistic method across various image sizes. With minimal impact on computation time as image size increases, the CCSE algorithm is highly efficient and well-suited for real-world

0009-0001-0854-8746

applications, particularly when dealing with high-resolution images. Besides, the CCSE algorithm exhibits excellent robustness as it maintains consistent performance across different parameter settings. This characteristic ensures that the algorithm can adapt to various scenarios and produce reliable results.

Overall, the CCSE algorithm outshines its counterparts in various key aspects, making it a promising choice for image segmentation tasks in a wide range of applications. Its runtime efficiency, robustness, and ability to produce visually and quantitatively accurate results demonstrate its potential for addressing the challenges faced in computer vision and image processing fields. This research provides valuable insights into the advantages of the CCSE algorithm and lays the foundation for further exploration and optimization of its capabilities.

REFERENCES

- [1] S. Na, L. Xumin and G. Yong, "Research on k-means Clustering Algorithm: An Improved k-means Clustering Algorithm," 2010 Third International Symposium on Intelligent Information Technology and Security Informatics, Jian, China, 2010, pp. 63-67, doi: 10.1109/IITSI.2010.74.
- [2] Umargono, Edy & Suseno, Jatmiko & Gunawan, S.K. (2020). K-Means Clustering Optimization Using the Elbow Method and Early Centroid Determination Based on Mean and Median Formula. 10.2991/assehr.k.201010.019.
- [3] El-Mandouh, Amira & A., Laila & A., Hamdi & H., Mohamed. (2019). Optimized K-Means Clustering Model based on Gap Statistic. International Journal of Advanced Computer Science and Applications. 10.10.14569/IJACSA.2019.0100124.
- [4] C. Connolly and T. Fleiss, "A study of efficiency and accuracy in the transformation from RGB to CIELAB color space," in IEEE Transactions on Image Processing, vol. 6, no. 7, pp. 1046-1048, July 1997, doi: 10.1109/83.597279.
- [5] Z. Zhang, W. Li and B. Li, "An Improving Technique of Color Histogram in Segmentation-based Image Retrieval," 2009 Fifth International Conference on Information Assurance and Security, Xi'an, China, 2009, pp. 381-384, doi: 10.1109/IAS.2009.156.
- [6] Gonzalez, Rafael C, et al. Digital Image Processing Using Matlab. Pearson/Prentice Hall, 2004, pp. 222-234.
- [7] Russ, J.C. (2011). The Image Processing Handbook (6th ed.). CRC Press. <https://doi.org/10.1201/b10720>