

# CS 169/268 Final Project

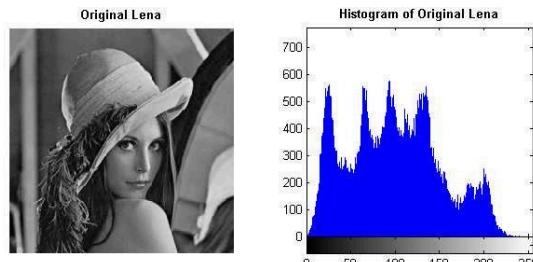
Group member: Jiabao Li, Qiuyu Ren, Xin Zhao, Jiawei Shi, Gaochang Zhao

## (a) Problem statement

This project focuses on the optimization of techniques for image segmentation. It specifically will look at implementing and comparing two approaches: threshold-based and similarity-based segmentation both through genetic algorithms and first-order alternating optimization. Threshold-based segmentation aims to find thresholds to separate gray-scale images into regions that give the maximum inter-class variance. In contrast, the similarity-based segmentation approach groups pixels into clusters by optimizing spatial and intensity similarities. The project also covers developing methods to assess the quality of segmented images and the convergence behavior of the algorithms. Comparing these techniques, the project tries to find out which is the best technique to achieve high-quality. Next we show the background definition and objective definition of each two approaches.

### Histogram

The histogram of an image is a graphical representation of the distribution of pixel intensity values in an image. It shows how frequently each pixel intensity value (or range of values) appears in the image, As shown below



The **probability mass function**  $P(i)$  represents the probability that a pixel in the image has intensity level  $i$ , defined as:

$$P(i) = \frac{n(i)}{N}, \text{ where } n(i) \text{ is the number of pixels in the image that have intensity } i,$$

$N$  is the total number of pixels in the image (i.e., the image size).

### Threshold-based

The multiple threshold segmentation aims to segment a single gray image into  $C$  groups (classes), where  $C > 2$ , using thresholds  $\mathbf{T} = \{T_1, T_2, \dots, T_{C-1}\}$ , the goal is to maximize the between-class variance, hence we define the objective function as

$$J(\mathbf{T}) = \sum_{k=1}^C P_k (\mu_k - \bar{\mu})^2$$

where  $P_k$  is the probability of  $k^{th}$  class, that is

$$P_k = \sum_{i=T_{k-1}}^{T_k} P(i)$$

$\mu_k$  is the mean value of gray scale in class  $k^{th}$ , that is

$$\mu_k = \frac{\mu_{T_k}}{P_k}, \quad \mu_{T_k} = \sum_{i=T_{k-1}}^{T_k} i \cdot P(i)$$

$\bar{\mu}$  is the mean value of the gray scale of the whole image, that is

$$\bar{\mu} = \sum_{i=0}^{255} i \cdot P(i)$$

We aim to optimize and obtain the thresholds  $\mathbf{T} = \{T_1, T_2, \dots, T_{C-1}\}$ , which shall also satisfy the constraint of  $T_1 < T_2 < \dots < T_{C-1}$ .

### Gradient descent optimization

Now we compute the partial derivative of  $J(\mathbf{T})$  with respect to  $T_k$ . First,

$$\frac{\partial P_k}{\partial T_k} = P(T_k), \quad \frac{\partial P_{k+1}}{\partial T_k} = -P(T_k)$$

$$\frac{\partial \mu_k}{\partial T_k} = \frac{\frac{\partial \mu_{T_k}}{\partial T_k} \cdot P_k - \mu_{T_k} \cdot \frac{\partial P_k}{\partial T_k}}{P_k^2} = \frac{T_k \cdot P(T_k) \cdot P_k - \mu_{T_k} \cdot P(T_k)}{P_k^2} = P(T_k) \frac{T_k \cdot P_k - \mu_k \cdot P_k}{P_k^2} = P(T_k) \frac{T_k - \mu_k}{P_k}, \text{ similarly}$$

$$\frac{\partial \mu_{k+1}}{\partial T_k} = P(T_k) \frac{\mu_{k+1} - T_k}{P_{k+1}},$$

then we have

$$\frac{\partial J}{\partial T_k} = \frac{\partial}{\partial T_k} [P_k (\mu_k - \bar{\mu})^2 + P_{k+1} (\mu_{k+1} - \bar{\mu})^2]$$

$$\begin{aligned}
&= \frac{\partial P_k}{\partial T_k} (\mu_k - \mu)^2 + P_k \frac{\partial(\mu_k - \mu)^2}{\partial T_k} + \frac{\partial P_{k+1}}{\partial T_k} (\mu_{k+1} - \mu)^2 + P_{k+1} \frac{\partial(\mu_{k+1} - \mu)^2}{\partial T_k} \\
&= P(T_k) (\mu_k - \mu)^2 + 2P_k (\mu_k - \mu) \cdot P(T_k) \frac{T_k - \mu_k}{P_k} + \\
&\quad [- P(T_k)] (\mu_{k+1} - \mu)^2 + 2P_{k+1} (\mu_{k+1} - \mu) P(T_k) \frac{\mu_{k+1} - T_k}{P_{k+1}} \\
&= P(T_k) (\mu_k - \mu) (2T_k - \mu_k - \mu) + P(T_k) (\mu_{k+1} - \mu) [- 2T_k + \mu_{k+1} + \mu]
\end{aligned}$$

thus we use gradient descent to optimize  $T_k$ , that is

$$T_k \leftarrow T_k - \eta \frac{\partial(-J)}{\partial T_k} = T_k + \eta \frac{\partial J}{\partial T_k}$$

## Genetic algorithm optimization

The chromosome is the encoded thresholds  $\mathbf{T} = \{T_1, T_2, \dots, T_{C-1}\}$ . The number of parents selected for mating in each generation is set to 5 to balance solution diversity and convergence speed. The number of solutions (individuals) in each generation's population is set to 20. The solver does the rest.

## Similarity-based

This idea is a transition from the threshold-based one, which can be thought of as a clustering of colors. However, normally the areas belonging to the same category in an image will not be scattered, thus the distance is considered, in other words, we increase the dimension of the sample points by two dimensions, namely their plane coordinates,  $(R, G, B) \rightarrow (R, G, B, x, y)$ .

We use the form of the k-means method, and the purpose is that in a single image, the pixels that have similar intensities (as well as distances) can be formed into the same group (segment area). To achieve this, we define the objective function as:

$$J = \sum_{i=1}^N \sum_{k=1}^K z_{ik} (\alpha ||\mathbf{x}_i - \boldsymbol{\mu}_k||^2 + \beta ||(r_i, c_i) - \mathbf{p}_k||^2)$$

where  $N$  is the total pixel number of the image,

$K$  is the cluster number,

$\mathbf{x}_i$  and  $\boldsymbol{\mu}_k$  are the pixel value of the current pixel  $i$  and the cluster center  $k$  respectively,

$(r_i, c_i)$  is the coordinate of pixel  $i$ ,

$\mathbf{p}_k$  is the coordinate of cluster center  $k$ ,

$z_{ik} \in \{0, 1\}$  is the indicator of whether the current pixel  $i$  belongs to cluster  $k$ ,

$\alpha$  and  $\beta$  are the weights.

We aim to minimize the objective function and then to get  $z_{ik}$ ,  $\mu_k$ , and  $\mathbf{p}_k$ .

### Gradient-based alternating optimization

To update the cluster center  $\mathbf{p}_k$ , we first denote  $d_{ik} = \|(r_i, c_i) - \mathbf{p}_k\|^2$ , then we have

$$\begin{aligned}\frac{\partial J}{\partial \mathbf{p}_k} &= \frac{\partial J}{\partial d_{ik}} \frac{\partial d_{ik}}{\partial \mathbf{p}_k} \\ &= -2\beta \sum_{i=1}^N z_{ik}[(r_i, c_i) - \mathbf{p}_k]\end{aligned}$$

set  $\frac{\partial J}{\partial \mathbf{p}_k} = 0$ , we have

$$\mathbf{p}_k = \frac{\sum_{i=1}^N z_{ik}(r_i, c_i)}{\sum_{i=1}^N z_{ik}}$$

similarly, we can update  $\mu_k$  by

$$\mu_k = \frac{\sum_{i=1}^N z_{ik} \mathbf{x}_i}{\sum_{i=1}^N z_{ik}}$$

finally,  $z_{ik}$  is updated following the rule

---


$$z_{ik} = \begin{cases} 1, & \text{if } k = \arg \min_j [\alpha \|\mathbf{x}_i - \mu_j\|^2 + \beta \|(r_i, c_i) - \mathbf{p}_j\|^2] \\ 0, & \text{otherwise.} \end{cases}$$


---

### Genetic algorithm optimization

The chromosome is the  $z_{ik}$ . The number of parents selected for mating in each generation is set to 5 to balance solution diversity and convergence speed. The number of solutions (individuals) in each generation's population is set to 20. The solver does the rest.

In this case, **GA is hard to optimize**, because the solution space is too large, we will **NOT** discuss the results of this part.

### (b) citations to previous work in the area

1. Basavaprasad, B., and S. Hegadi Ravindra. "A survey on traditional and graph theoretical techniques for image segmentation." *Int. J. Comput. Appl* 975 (2014): 8887.
2. Liu, Xiaolong, Zhidong Deng, and Yuhang Yang. "Recent progress in semantic image segmentation." *Artificial Intelligence Review* 52 (2019): 1089-1106.
3. Zhang, Hui, Jason E. Fritts, and Sally A. Goldman. "Image segmentation evaluation: A survey of unsupervised methods." *computer vision and image understanding* 110.2 (2008): 260-280.
4. Webpage [How to do image segmentation without machine learning - Quora](#)
5. The dataset we are using is from [Berkeley Segmentation Dataset 500 \(BSDS500\)](#) since it is a coarse-grained segmentation dataset, which is very friendly to this type of experimental project. It contains 500 natural images, and each image was segmented by five different subjects on average. Some samples are shown below.



(c) your decomposition of the project into separate portions including all responsibilities and credits by group member, pointing external code used in each portion, and any milestones or checkpoints you used;

#### **Project Decomposition:**

The project was divided into five major components, each with specific tasks and objectives:

1. Analyze the problem and determine optimization methods; Main Contribution:  
*Jiabao Li Qiuyu Ren Shi Jiawei*
2. Investigating relevant work, looking for datasets and evaluation metrics, etc.  
Main Contribution: *Gaochang Zhao Xin Zhao Shi Jiawei*
3. Derive the objective functions to get the derivatives and closed-form solutions;  
Main Contribution: *Jiabao Li Qiuyu Ren*
4. Code implementation and analysis on image properties. Main Contribution:  
*Jiabao Li Shi Jiawei Xin Zhao Gaochang Zhao Qiuyu Ren*
5. Analyze the results and organize the project report. Main Contribution:  
*Gaochang Zhao Xin Zhao Jiabao Li*

Image processing requires a lot of background knowledge. Each team member has a different level of background knowledge, but we are all passionate about this project and actively completed this task. In the process, we learned and solved problems, so credits for each group member is 20%.

(d) your experience in coding it up;

The code is well-structured and modular, with the two methodologies and four solutions organized into different sections. This structure, combined with clear comments and Markdown titles, made implementation intuitive and easy to debug. However, during the implementation of the algorithms, there were some challenges, especially in defining effective fitness functions in GA and in gradient calculations in the gradient descent method. With respect to similarity-based clustering, the balancing of parameters, like  $\alpha$  and  $\beta$ , demanded several experiments toward an optimum.

Code debugging and optimization included resolving the deprecation of `get_cmap` in Matplotlib, as well as computational efficiency improvement with NumPy for batch

operations. In the genetic algorithm, improvements were made on the parameter tuning, such as initialization and mutation strategies to help the convergence of results. Outputs were presented with visualizations using Matplotlib and OpenCV. It provided clear insight into the difference between different algorithms and the impact of parameters; the segmentation images helped in receiving valuable feedback from iterative improvements.

Working with PyGAD, OpenCV, and Matplotlib was a nice experience, enhancing practical knowledge with respect to image data handling and optimization techniques. This overall process has provided deeper understanding related to image segmentation methodologies and their trade-offs-both technical and conceptual.

(e) your experience and results including quantitative results in testing it, with relevant quantitative comparisons

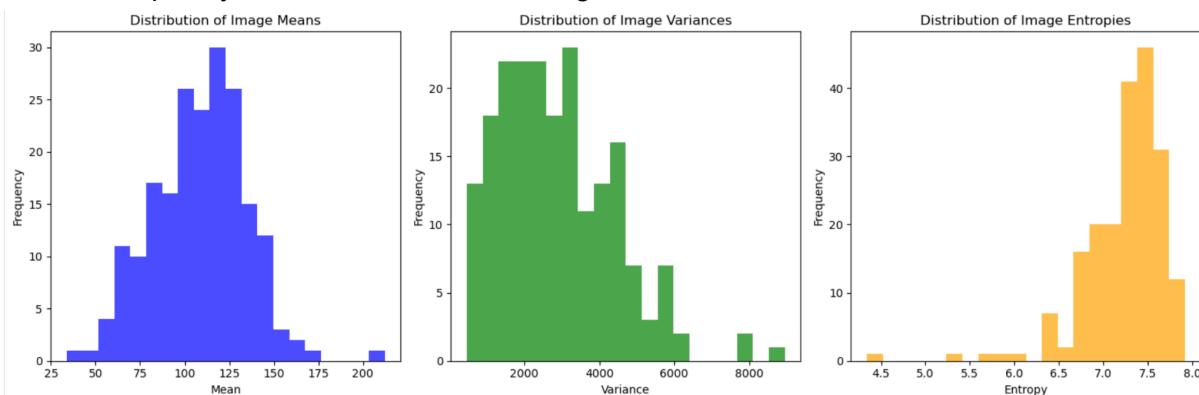
## Image Analysis

Our experiments are conducted under grayscale images, which is converted by:

$$Gray = R * 0.299 + G * 0.587 + B * 0.114$$

Then we analyze grayscale images by computing their statistical properties: mean, variance, and entropy, and visualize the distribution of these measures by means of histograms:

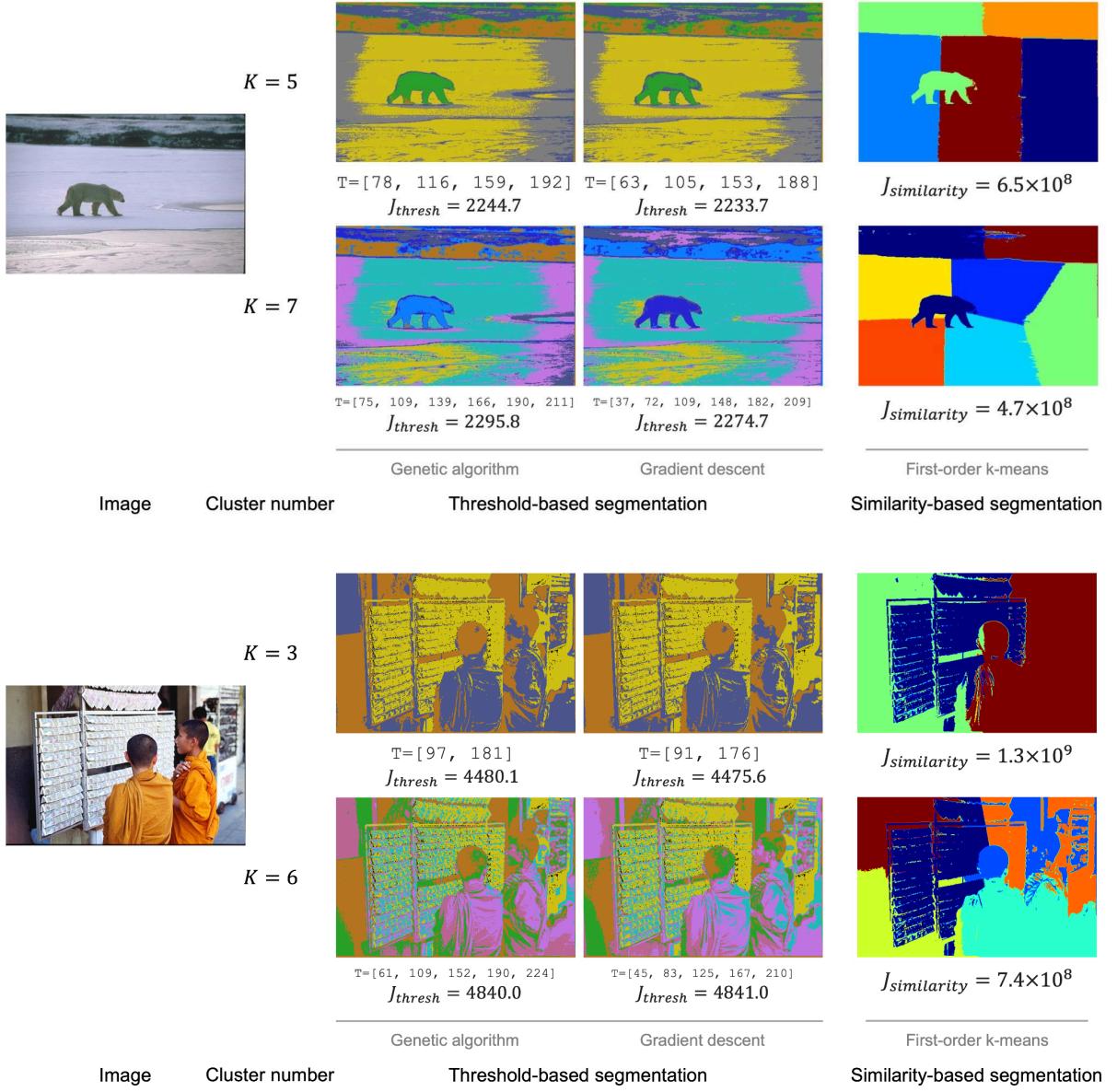
- *Mean*: describes the average level of overall brightness of the image.
- *Variance*: It measures the spread of image pixel values, that is, the degree of change in brightness.
- *Entropy*: Measures the amount of pixel information of an image, reflecting the complexity or randomness of the image.



We observed from experiments that images with larger contrast usually achieve better segmentation quality using our method, and these images are usually distributed in the right part of the variance and entropy maps (green and yellow above).

## Segmentation result

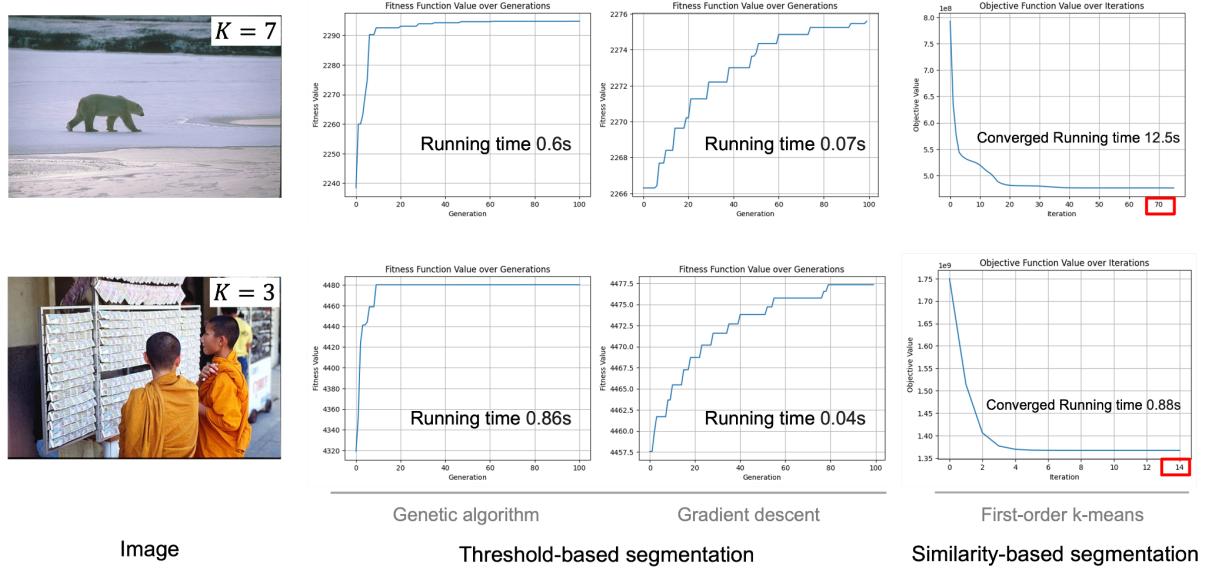
We show three types of segmentation results on two images below, note that the larger  $J_{thresh}$ , the better; the lower  $J_{similarity}$ , the better. We can find that the results shown by the threshold-based method and the similarity-based method are very different. This is because in the similarity-based method, we also consider spatial similarity, making its results less likely to be dispersed. And among the threshold-based methods, the results using the global optimization algorithm are numerically better.



## Evaluation analysis

We show below the performance metrics for two images with different numbers of clusters. Although the two threshold-based optimization methods have similar performance, their running time differs by nearly 10 times. The gradient-based

method is very fast. Since the similarity-based method uses the same iterative strategy based on k-means, its running time will vary depending on the convergence speed. In our actual experiments, this method usually takes more time than the previous two methods.



(f) conclusions including evaluation of your approach including the optimization methods and application domain aspects of the project;

## Conclusions

This project compared Threshold-Based and Similarity-Based for image segmentation, we utilized GA and gradient descent optimization approaches in the threshold-based method, and only first-order alternating optimization in the similarity-based method. All three approaches have particular merits. GAs have performed well in determining optimum thresholds for segmentation by searching within the solution space, while this method needs more care in selecting its parameters and requires a trade-off between runtime and performance, because generally speaking, more populations and iterations mean longer runtimes. Gradient Descent can be computationally cheap, with fast runtime for applications in real time; however, it gets easily trapped into local optima when the scenario becomes complicated. The first-order alternating update for similarity-based method combined color and spatial information to achieve perceptually aligned results at the cost of higher computational demands and sensitivity to parameters.

In summary, each of these methods has pros and cons. GAs are suitable for global optimization, Gradient Descent is good for real-time tasks, and the similarity-based method that considers the spatial information is consistent with human perception.

## **Application domain aspects**

Image segmentation plays a crucial role in numerous fields due to its ability to isolate and identify objects or regions of interest in an image. Though our approach in this project was straightforward, the significance of image segmentation in real-world applications cannot be overstated. It is fundamental in medical imaging, enabling the identification of tumors, organ boundaries, and anomalies for diagnosis and treatment planning. In autonomous driving, segmentation helps vehicles understand their surroundings by differentiating between roads, pedestrians, and obstacles. Other applications include satellite image analysis for land use classification, agricultural monitoring, and augmented reality for object recognition and interaction.

(g) a bibliography including any code sources or web sources you relied on and anything else you cited in (b)

1. Gad, Ahmed Fawzy. "5 Genetic Algorithm Applications Using Pygad." DigitalOcean, September 23, 2024.  
<https://www.digitalocean.com/community/tutorials/genetic-algorithm-applications-using-pygad>
2. "python中的ga包怎么用" , December 1, 2024.  
[https://blog.51cto.com/u\\_16175476/12707173](https://blog.51cto.com/u_16175476/12707173)
3. AKBASLI, IZZET TURKALP. "Genetic Algorithm: Tutorial of Pygad." Kaggle, July 20, 2022.  
<https://www.kaggle.com/code/zzettrkalpakbal/genetic-algorithm-tutorial-of-pygad>