CS484/555 - Introduction to Computer Vision
Homework 3
Report

Efe Tarhan
22002840
Electrical and Electronics Engineering

Content

I) Introduction **2**

SLIC (Simple Linear Iterative Clustering) Algorithm *2*

Gabor Filtering *3*

K-Means Clustering *4*

II) Part 1. ..... **5**

III) Part 2. ..... **8**

IV) Part 3. ..... **12**

V) Part 4. ..... **14**

Conclusion **18**

References **18**

I) Introduction

SLIC (Simple Linear Iterative Clustering) Algorithm

Simple linear iterative clustering (SLIC) algorithm is a method for generating superpixels which are perceptually meaningful atomic regions in images [1]. The method employs an adapted version of the classical k-means algorithm. The algorithm initializes the cluster centers uniformly on the 2D pixel grid using a separation parameter $S = \sqrt{N_p/N_{SP}}$ where N_p is the number of pixels and N_{SP} is the number of desired superpixels. After initializing the cluster centers separated each other from S pixels, the centers are shifted towards the direction with the lowest gradient in the 3x3 neighborhood to eliminate the chance of initializing a superpixel on an edge.

After the initialization part the algorithm works like k-means clustering which assigns a cluster membership for each pixel in an iterative manner but in a limited search space which is shown in Figure 1. The algorithm uses a weighted distance measure D which applies different weights on the parameters related to colors and locations to make sure desired number of superpixels or image size does not affect the algorithm. The compactness parameter of the function determines the weights that will be assigned for color and distance measures which changes the shape of the superpixels.

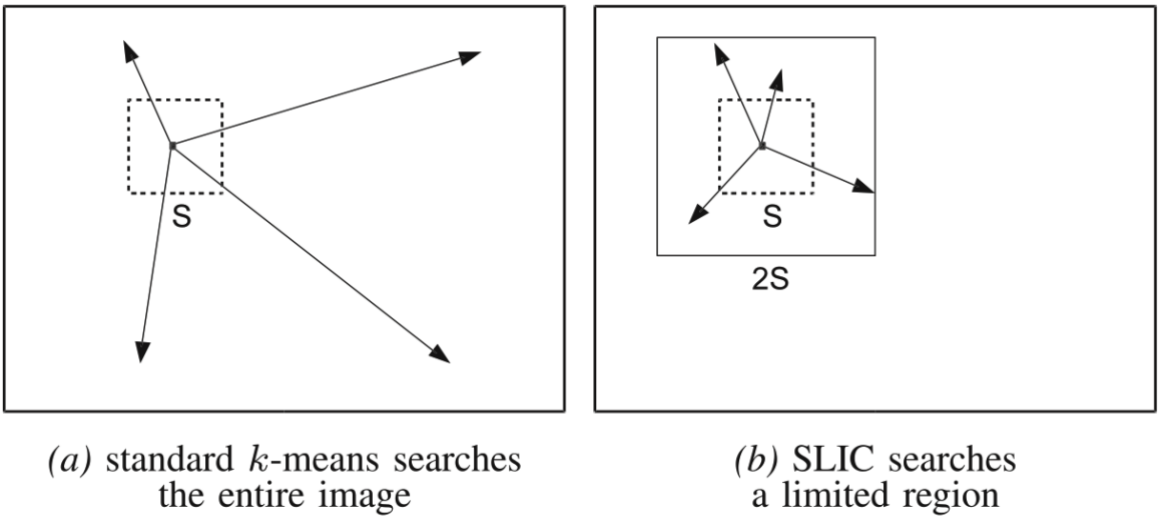

Fig. 1 The k-means clustering algorithm searches the whole space (a) while the SLIC algorithm only operates for pixels in a defined neighborhood around the pixel (b).

For the homework, the superpixels are obtained by using the custom SLIC function developed for MATLAB environment by the IVRL in EPFL [2]

Gabor Filtering

Gabor filter is a linear 2D filter consisting of a sinusoidal with a gaussian envelope. The mathematical expression of the filter can be seen below.

$$g(x, y; \lambda, \theta, \psi, \sigma, \gamma) = \exp\left(-\frac{x'^2 + \gamma^2 y'^2}{2\sigma^2}\right)\cos\left(\frac{2\pi x'}{\lambda} + \psi\right)$$

where the transformations for rotation are:

$$x' = x\cos\theta + y\sin\theta$$
$$y' = -x\sin\theta + y\cos\theta$$

The additional parameters are also as the following:

- λ is the wavelength of the sinusoidal which influences the scale of the features that the filter will respond to.
- θ is the orientation parameter which determines the normal of the sinusoidal stripes of the Gabor function.
- ψ is the phase or offset of the Gabor filter that determines the area that the features will be created for the given function.
- σ is the standard deviation of the Gaussian envelope that determines the smoothness of the image and ability of the filter to capture finer details.
- γ is the value that determines the ellipticity of the filter where smaller γ values result in sharper ellipses.

Gabor feature for each pixel of the image can be obtained by convolving Gabor filter with the image. A set of 2D Gabor filters generated using the default MATLAB parameters by only changing the λ and θ can be seen in Figure 2.
For the homework, Gabor filters for desired values of λ and θ are obtained using the "**gabor**" [3] and "**imgaborfilt**" [4] functions of MATLAB. The resultant filtered images are concatenated on top of each other which will create a new pixel consisting of concatenated 16 pixels which will be used as the feature vector for the given pixel.

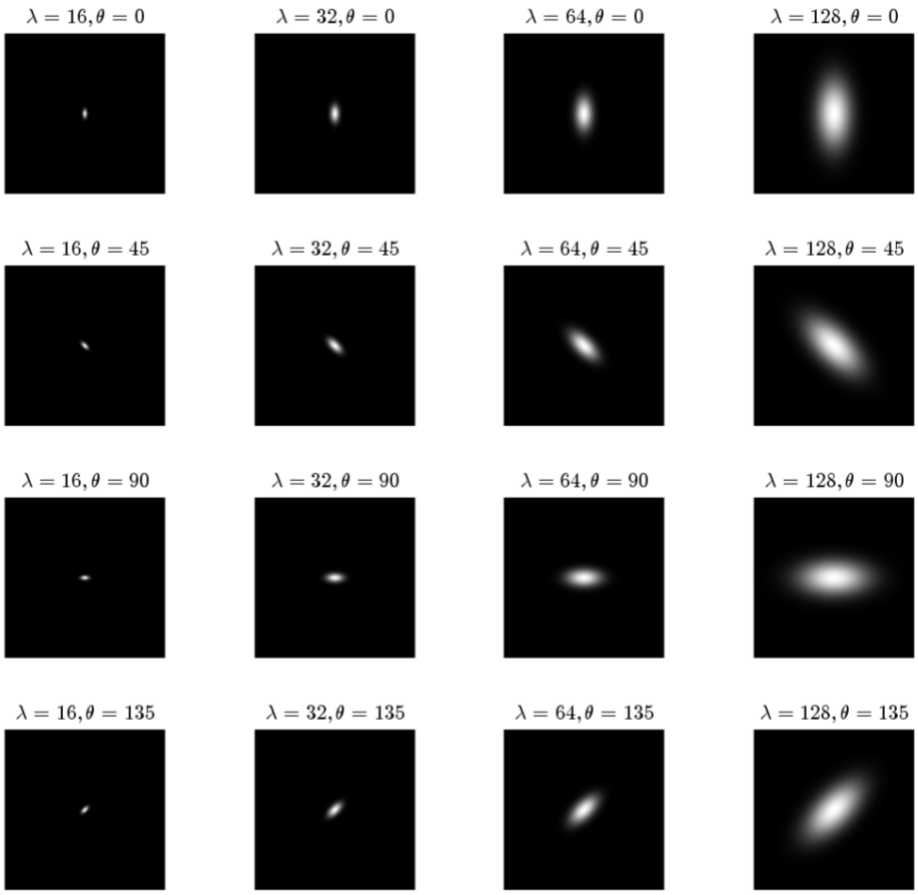

Fig. 2 Gabor filters for different wavelength (λ) and orientation (θ) parameters.

K-Means Clustering

K-means clustering is one of the basic and classical machine learning techniques available in the literature to cluster data into k classes. It is considered as an unsupervised learning technique because it does not require any labels for the data. Also, this technique does not require any predefined model parameters to be learned like in the deep learning, so it is also called as a model-free or non-parametric technique. For this assignment Euclidian distance is preferred as the metric of k-means clustering algorithm. Then the loss function of the algorithm can be given as:

$$J = \sum_{i=1}^{N} \sum_{j=1}^{K} I(x_i \in c_j)(x_i - \mu_j)^2$$

where μ_j is the center of the cluster c_j and $I(.)$ is the indicator function that gives 1 if the given data belongs to the current cluster. It is hard to given optimization task for all dataset at once since it requires the optimization of many parameters at once. Rather, an iterative algorithm is applied with the given steps:

Expectation:

$$I(x_i \in c_j) = \begin{cases} 1, & c_{\underset{m}{\operatorname{argmin}}(x_i-\mu_m)^2 \ \ m\in\{1,2,\dots,k\}} \\ 0, & else \end{cases}$$

Maximization:

$$\mu_j = \frac{\sum_{i=1}^{N} I(x_i \in c_j) x_i}{\sum_{i=1}^{N} I(x_i \in c_j)}$$

From this perspective it is guaranteed that the algorithm will converge to a local minimum point after required number of iterations. The K-means clustering algorithm will be used for grouping the pixels into clusters to segment the image into regions using pixels with similar features.

II) Part 1.

For the first part of the homework, the SLIC algorithm has been tested using the code mentioned in the introduction part of the report. There are 2 different important hyperparameters of the SLIC algorithm as mentioned in the introduction part. First parameter is the number of superpixels N_{SP} which determines the initial number of cluster centers for algorithm to be started, this parameter automatically determines the average size of the superpixels. Another parameter of SLIC is the compactness which is referred as “m” in the original paper but will be mentioned as “C” in this report. Compactness is the weight of the spatial difference for the distance measure for k-means algorithm. If the value of compactness parameter is high the algorithm has more tendency to penalize clustering points far away from the centers and therefore the

superpixels are generated with a tighter shape that fits to the image boundaries. The results of evaluating the algorithm with different number of superpixels to be obtained and a different value for compactness is shown in the Figures 3, 4, 5 and 6. Superpixel boundaries are highlighted with red colored lines using the MATLAB's "**edge**" function [5].

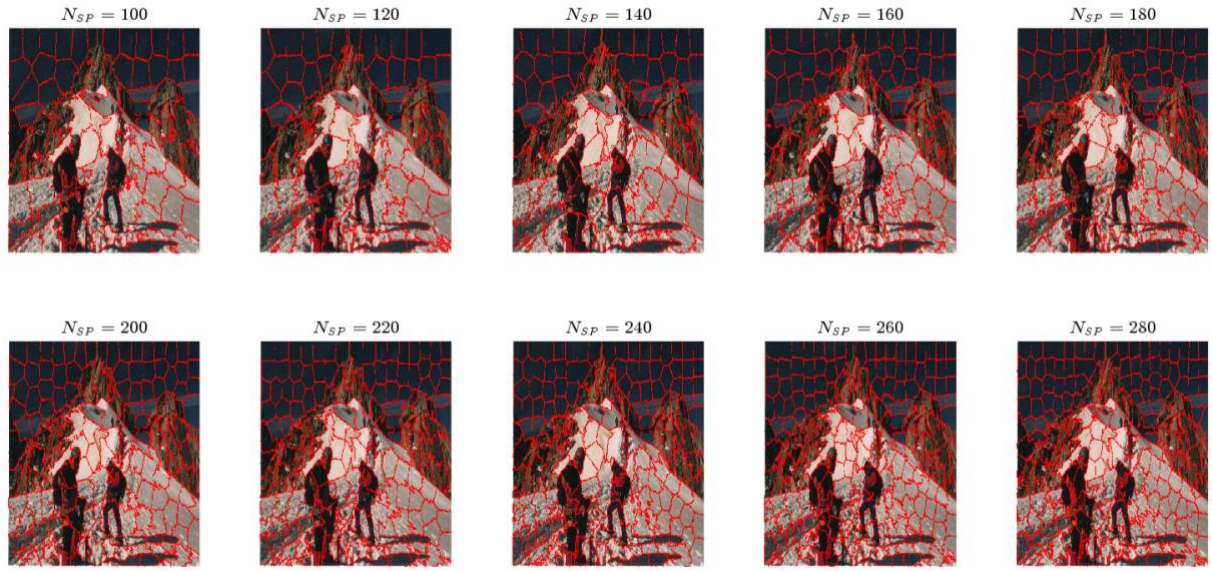

Fig. 3 Effect of changing the number of superpixels (N_{SP}) for the image "3.jpg".

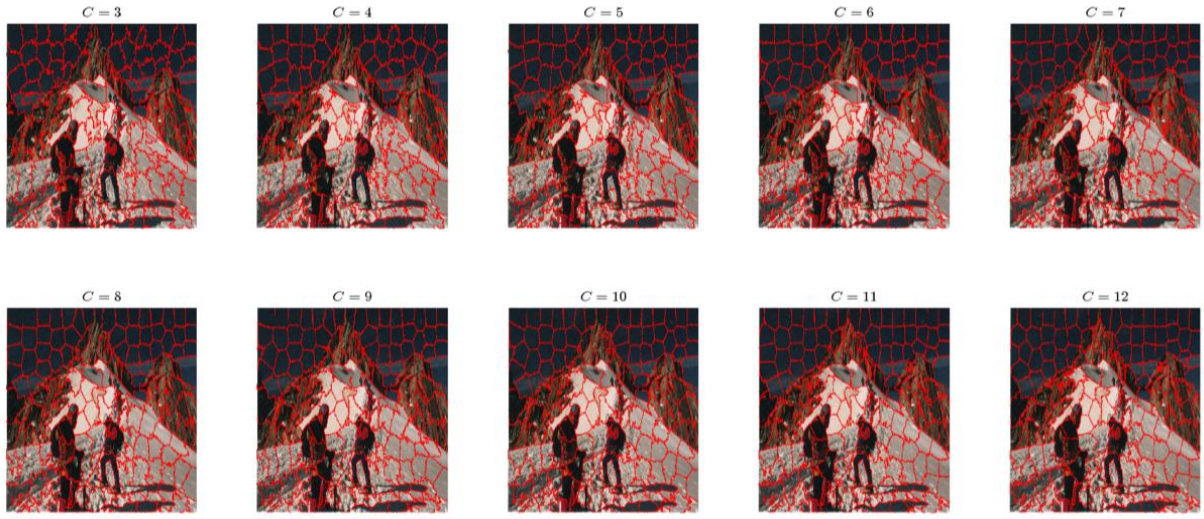

Fig. 4 Effect of changing the compactness parameter (C) for the image "3.jpg".

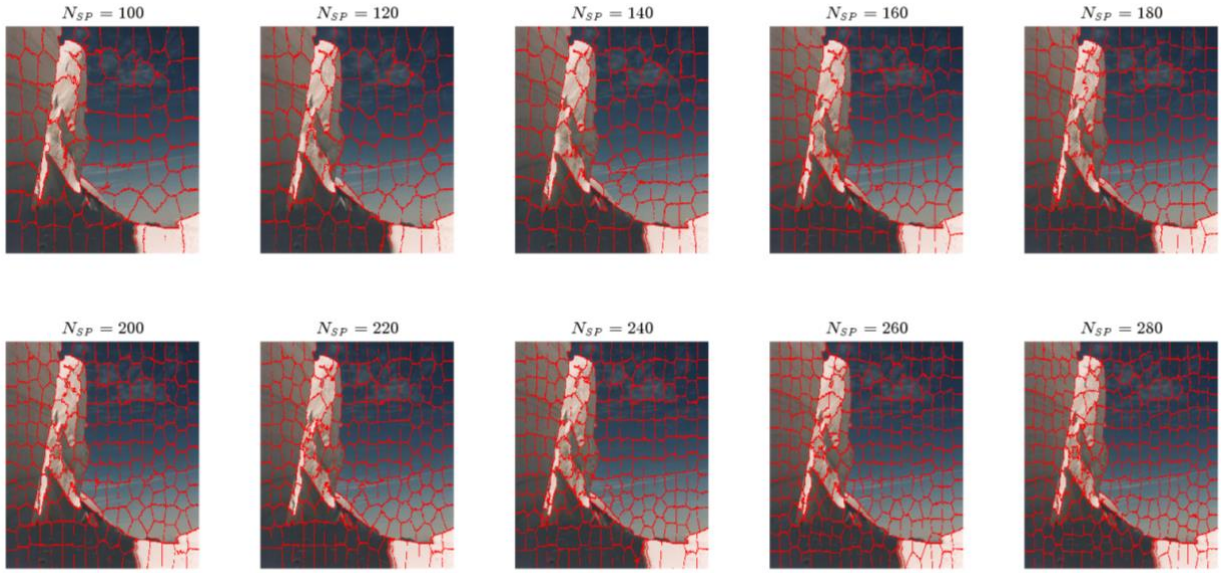

Fig. 5 Effect of changing the number of superpixels (N_SP) for the image "8.jpg".

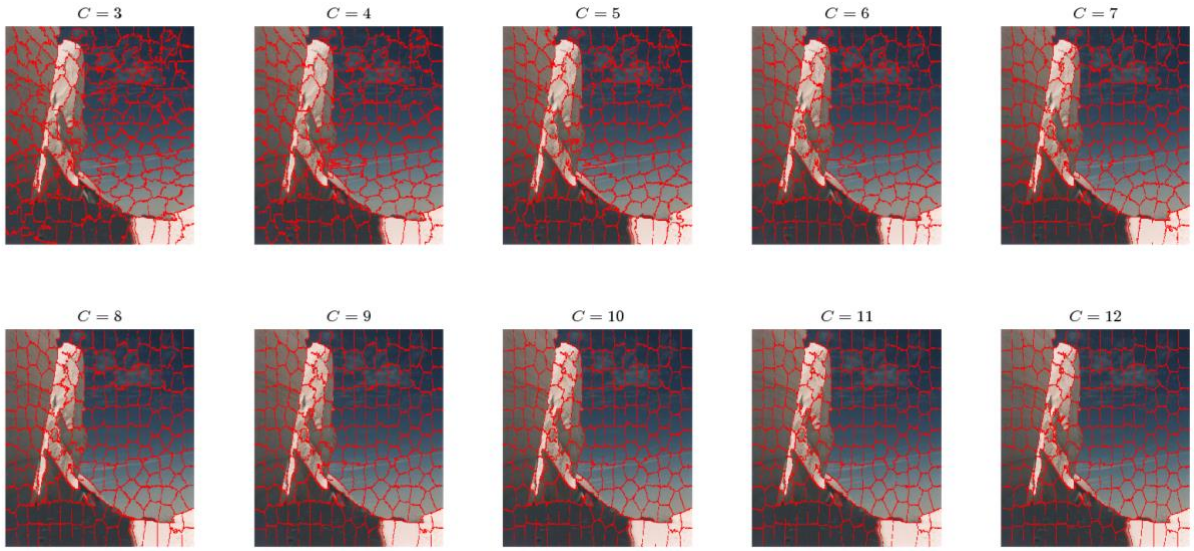

Fig. 6 Effect of changing the compactness parameter (C) for the image "8.jpg".

By observing these parameter sets and effect of these parameters on images with different level of detail. Also, the effect of selecting various parameters for the number of superpixels and compactness parameter is considered. From the Figure decreasing the compactness parameter increases the superpixels adaptation to the detailed boundaries because of the role of the color distance. In contrast from Figure 6, the over-adaptation of the superpixels to the colors and textures in the image resulted in overfitted superpixels which might not represent the true effect of the selected region

but only the color property of it. Therefore, the value of the compactness causes the superpixels to over-adapt or under-adapt to the textures in the image if it is selected too small or high. Therefore, the value of compactness has been selected as 10. By observing all results, the desired number of superpixels is selected as 200 which is mostly connected with the latter stages of the assignment. Segmentation results for all images with the selected parameters can be seen below.

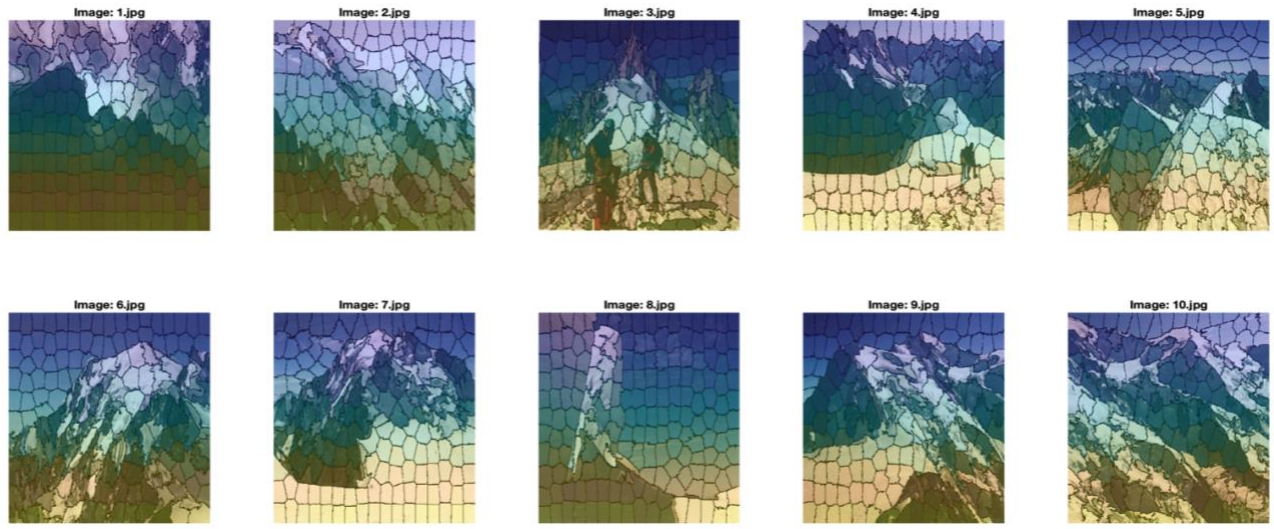

Fig. 7 Result of applying the SLIC algorithm to all images in the dataset using $N_{SP} = 200$ and $C = 10$. Each superpixel is represented with a different color.

III) Part 2.

In this part Gabor filtering is applied to the images with generated superpixels. The details of the parameters of the Gabor filter are described in the introduction part. For this assignment the wavelength (λ) and the orientation (θ) parameters of the MATLAB's "**gabor**" function is used which creates a filter bank. Different parameter sets are applied on different images, result of the experiments are shown in Figures 8, 9, 10, 11 where different filters with specified parameters are tested on 4 different images. The parameter λ has been changed over the values $[2, 4, 6, 8, 16, 32, 64, 128]$ and the θ parameter has taken values $[0°, 15°, 30°, 45°, 60°, 75°, 90°, 105°, 120°, 135°, 150°, 165°]$.

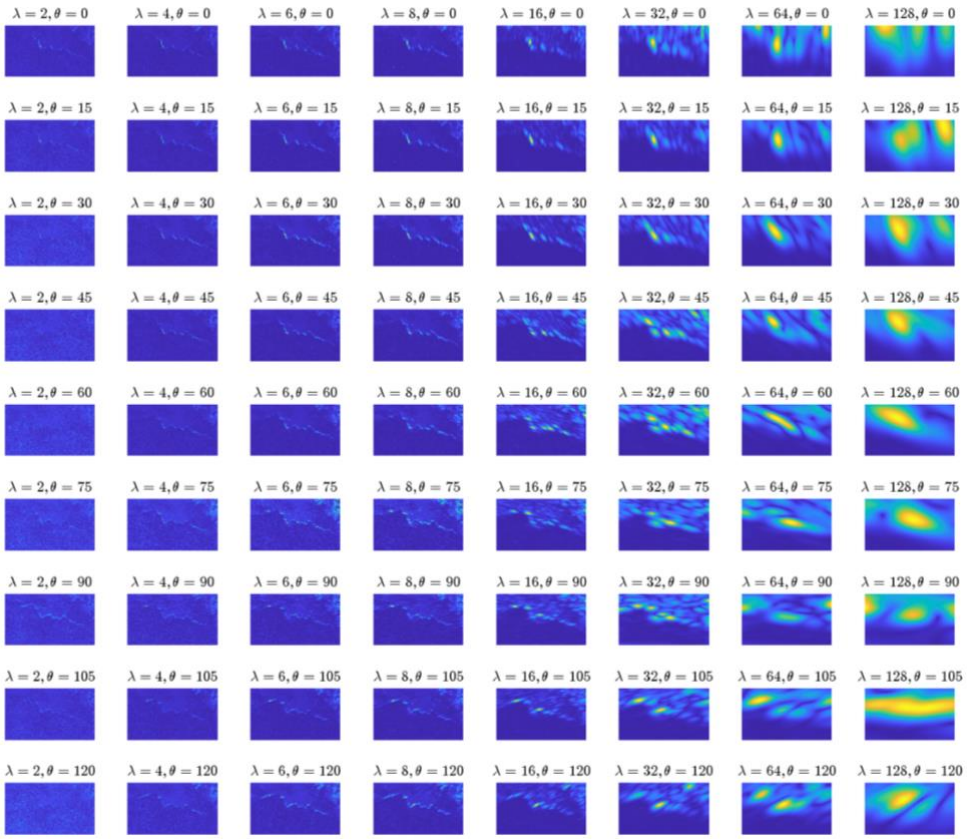

Fig. 8 Gabor features of image "1.jpg" for different parameters

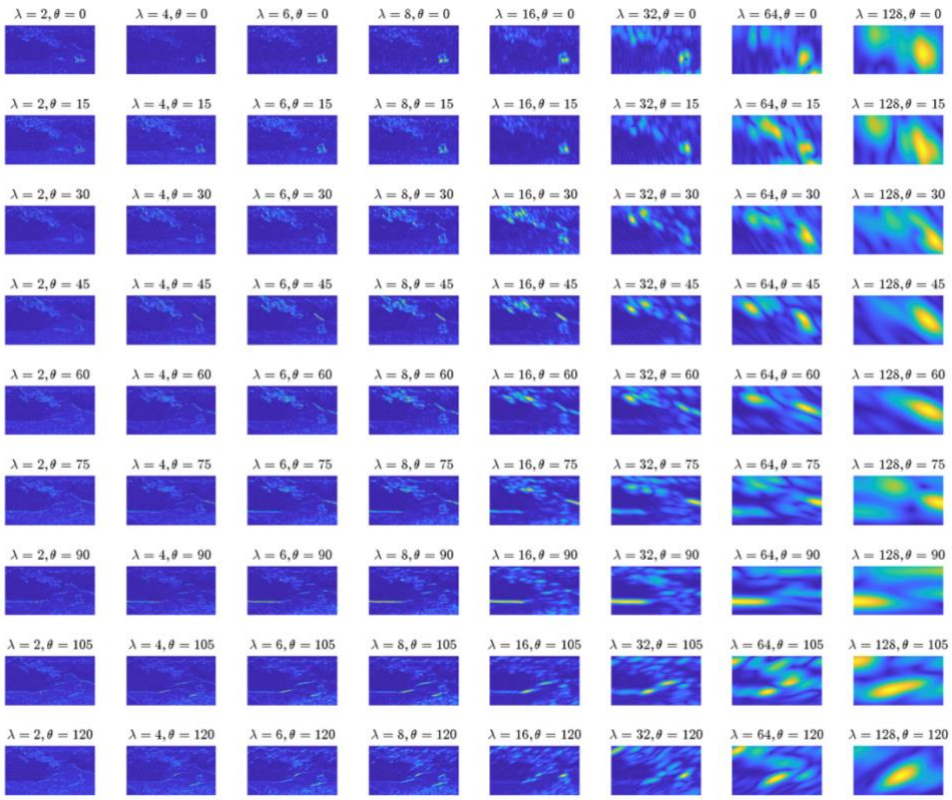

Fig. 9 Gabor features of image "4.jpg" for different parameters

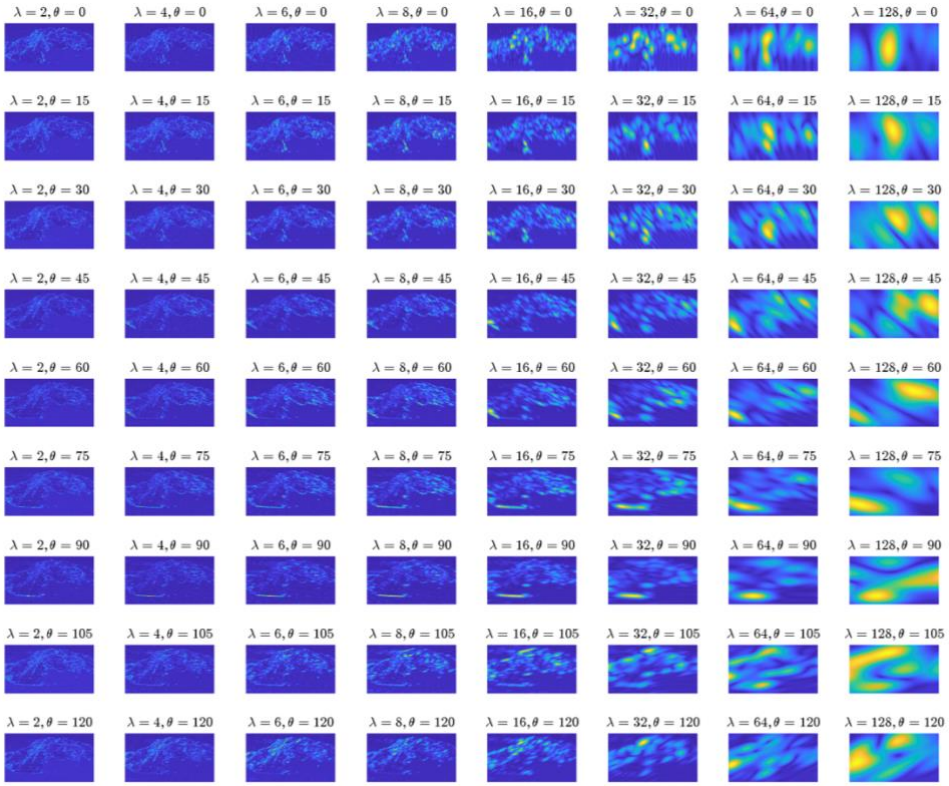

Fig. 10 Gabor features of image "7.jpg" for different parameters

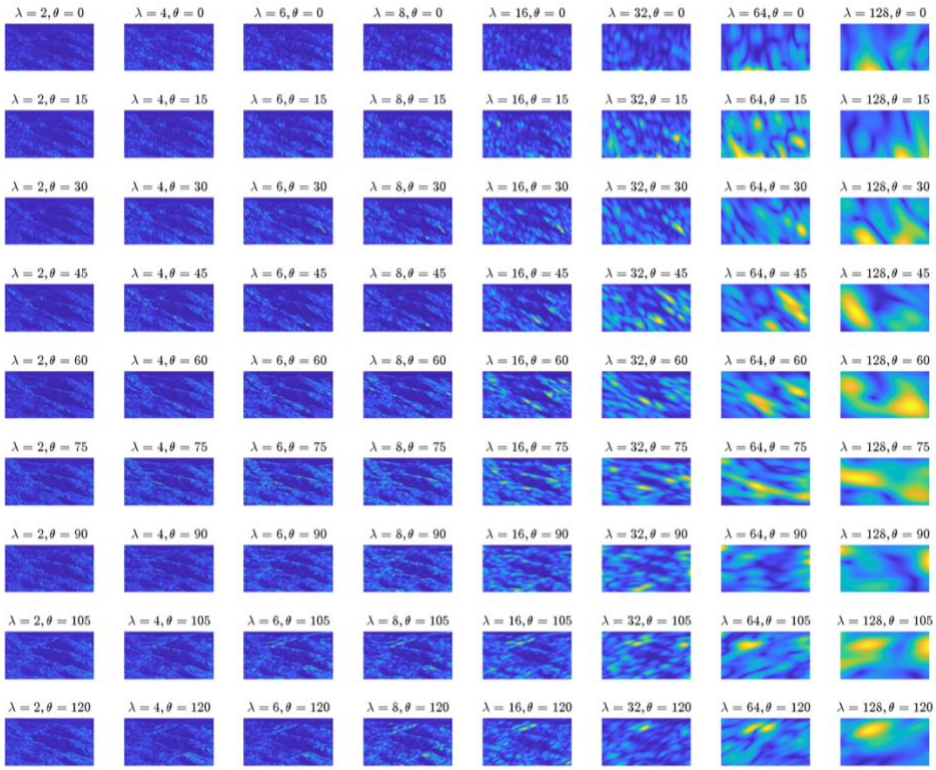

Fig. 11 Gabor features of image "10.jpg" for different parameters

By observing the results, it can be seen that increasing the scale parameter to values 16, 32 and 64 causes severe loss of details which might confuse the classification

algorithm although it captures coarser details. Therefore, the candidate wavelets were $[2, 4, 6, 8]$ or $[2, 4, 8, 16]$. By visual inspection of the results the decision was made towards using the set $\lambda \in [2, 4, 8, 16]$ because of better generalization performance. Result of several images with the selected parameters are shown in Figures 12, 13 and 14.

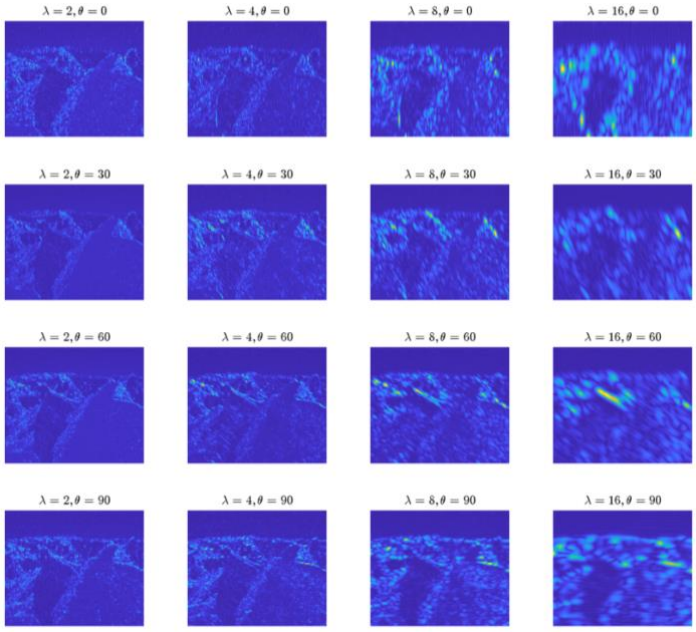

Fig. 12 Finalized Gabor Features of the image "2.jpg".

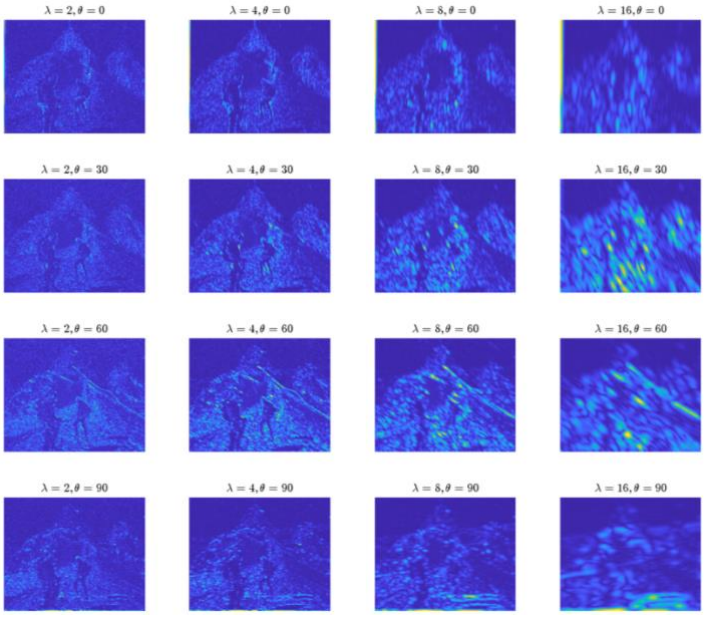

Fig. 13 Finalized Gabor features of the image "3.png"

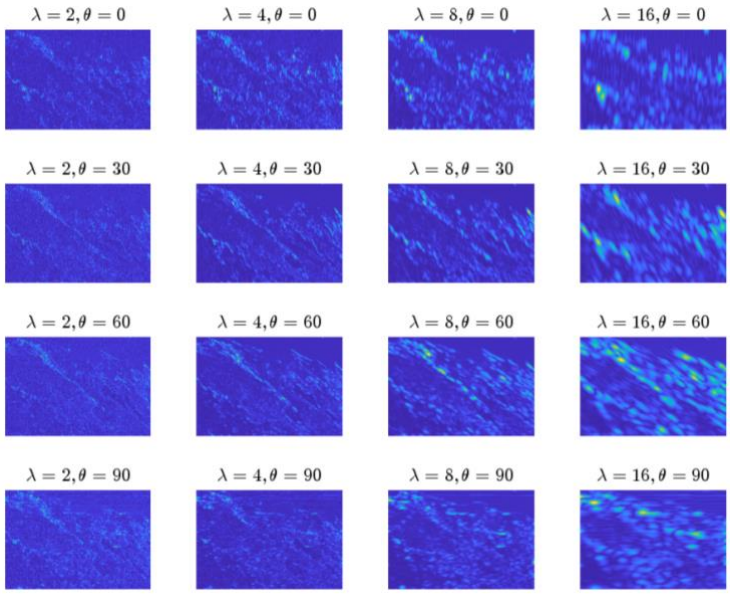

Fig. 14 Finalized Gabor features of the image "5.jpg"

IV) Part 3.

After extracting the features each pixel became a vector in R^{16}. Following that the average of the features of the pixels that belong to the same superpixel are calculated and the new average features are assigned to the superpixels. Following that the feature vectors of each superpixel are gathered and feed into the k-means clustering algorithm for different values of k. The results for testing the algorithm using different values of k can be seen below.

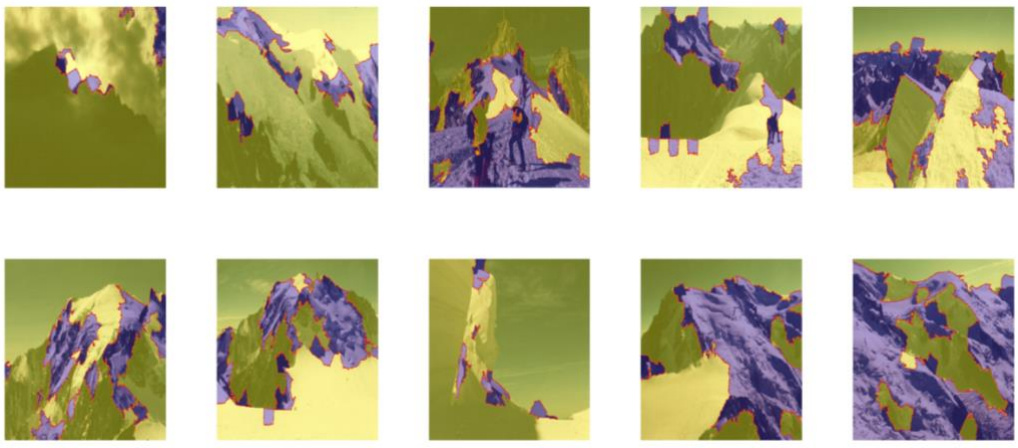

Fig. 15 K-means clustering result for k = 2.

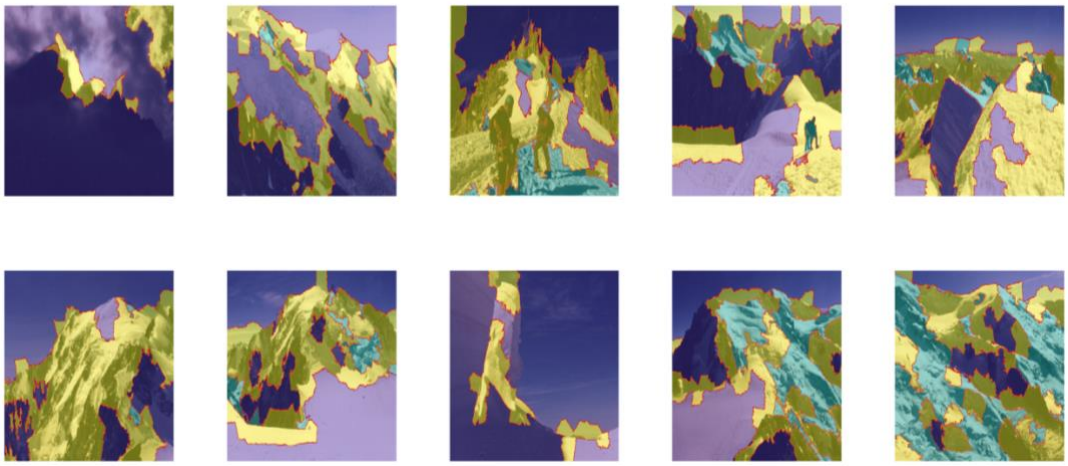

Fig. 16 K-means clustering result for k = 3.

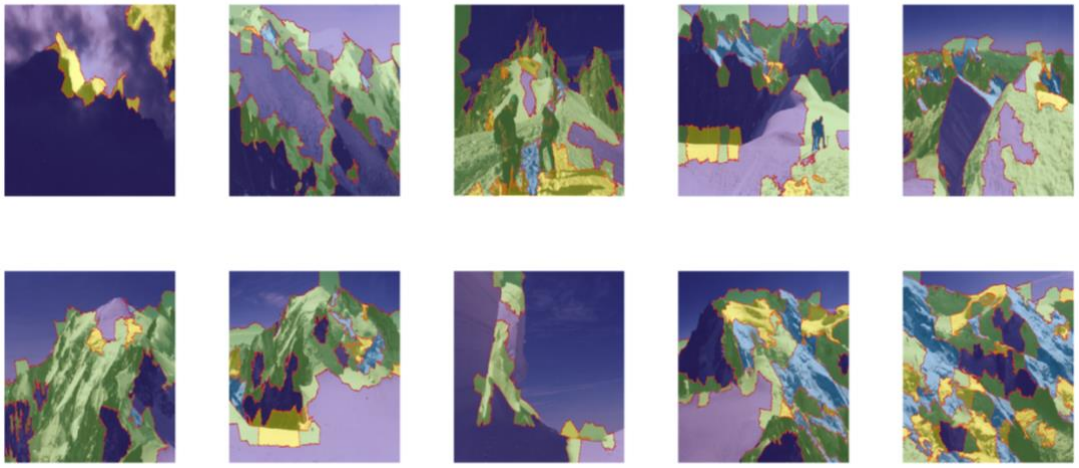

Fig. 17 K-means clustering result for k = 4.

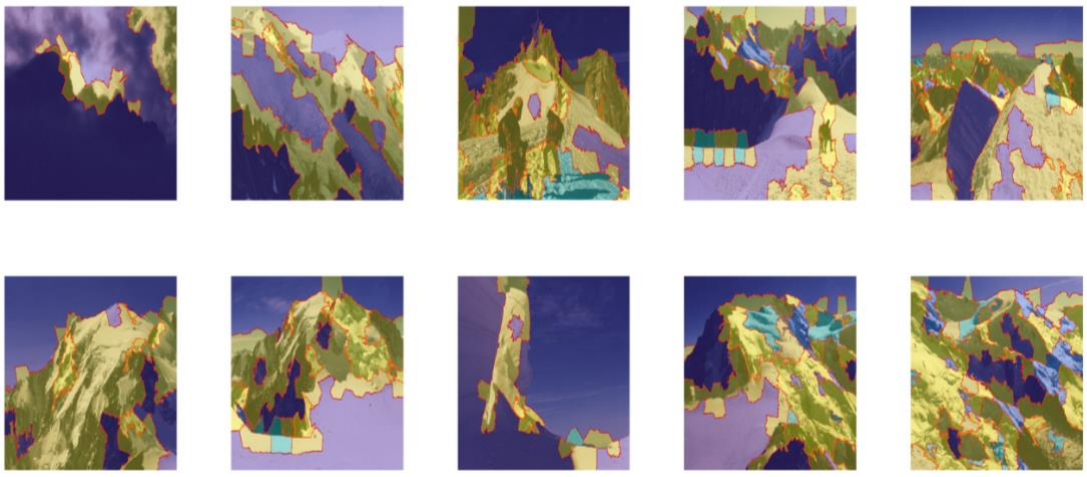

Fig. 18 K-means clustering result for k = 5.

By observing the results in Figures 15, 16, 17 and 18, it can be argued that there are 3 major patterns or scenes in most of the images. The sky, mountains, and the objects on the mountains are 3 types of textures in the images. Also, by inspecting the success of the algorithm for different values of k the segmentation task is most successfully completed by using 3 initial clustering centers. Results of all images using 3 means clustering algorithm is shown below in Figure 19.

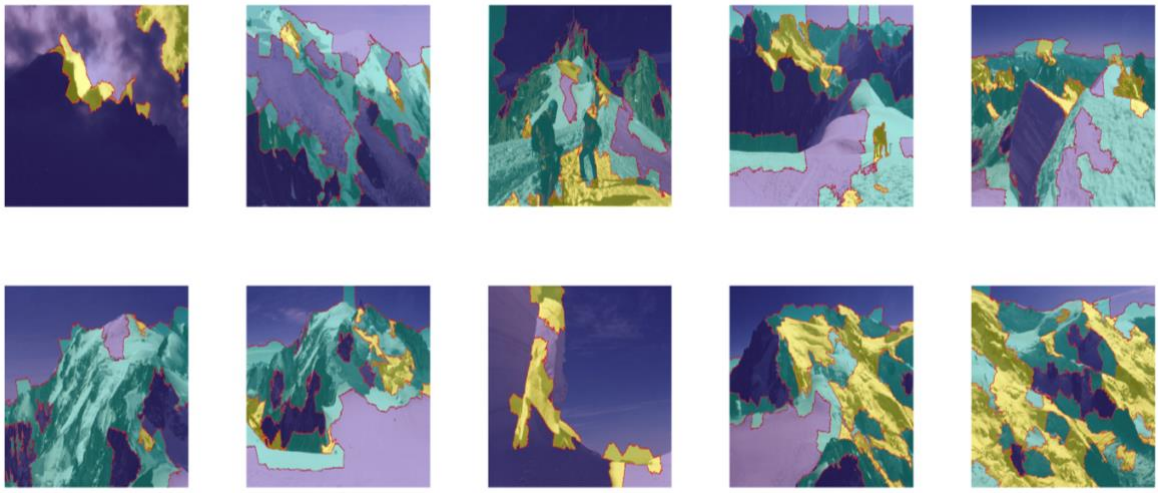

Fig. 19 Results of K-Means clustering algorithm applied using Gabor features.

V) Part 4.

In this part the Gabor features obtained in Part 2 are extended by introducing a neighborhood concept that includes the average of the feature of the neighbors are concatenated to the initial feature vector of the clusters. Therefore, the features have become more aware of the features of the surrounding superpixels. The neighboring conditions can be given as the following and can be seen from Figure 20:

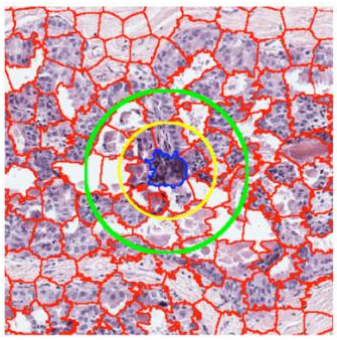

Fig. 20 Example of the neighborhood regions.

The mathematical condition for a superpixel x to be in the first level neighborhood $(N1)$ of a superpixel x_{SP} is that the superpixel must contain a number of its pixels inside a hypothetical circle with radius 1.5 times of the defined radius of the superpixel. On the other hand, the mathematical condition for a superpixel x to be in the second level neighborhood $(N2)$ of a superpixel x_{SP} is that the superpixel must contain a number of its pixels inside a hypothetical ring with inner radius 1.5 times and outer radius 2.5 times of the defined radius of the superpixel.

The first threshold is the ratio of the superpixels to be appeared in each region for assigning a neighborhood membership. The second parameter is the radii of the superpixels. But this parameter can be approximated by the following equality:

$$\tilde{r} = \sqrt{\frac{1}{\pi}\frac{\#Total\ Pixels}{\#Super\ pixels}}$$

As an example, for if there are 200 superpixels the approximate radius is equal to nearly 25 pixels. If this value is compared with the actual results the pixel radius for a superpixel with comparably uniform shape appears to be 30 which is close to our approximation. This calculation is shown in Figure 21.

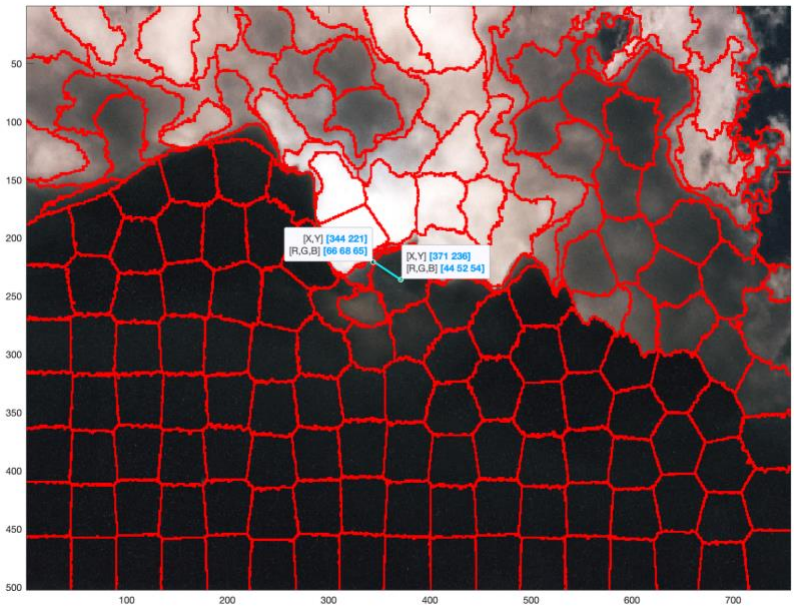

Fig. 21 Calculated radius of a superpixel.

But because of determining a hard threshold causes some large superpixels to not have any neighbors and therefore generates a value absence problem, different values for the

radius parameter will still be tested. In Figure 22. the threshold for the minimum number of pixels for it to be a member of a superpixel neighborhood is swept from 100 to 1000 with a step of 100.

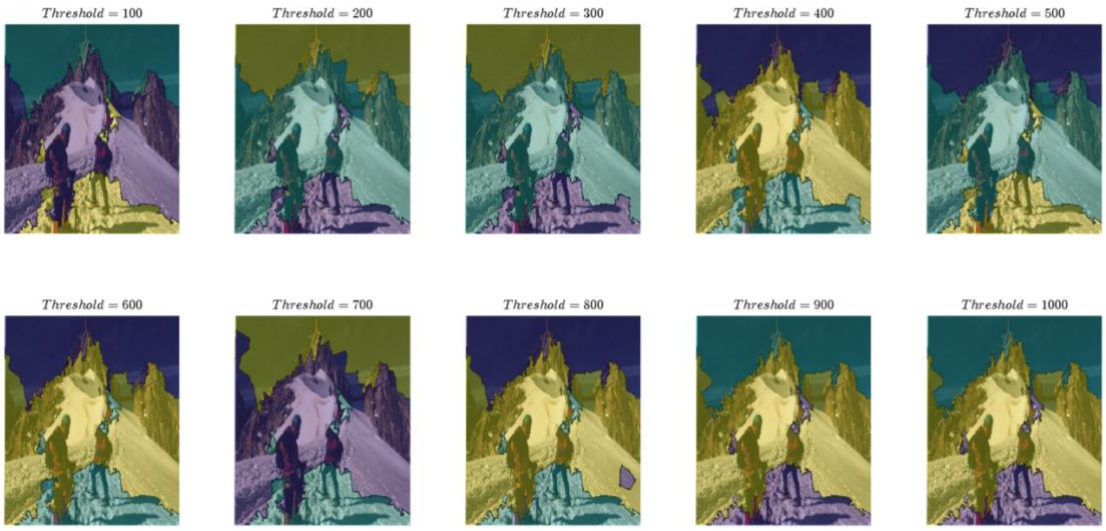

Fig. 22 Clusters generated using different membership threshold values.

As it is mentioned before the cluster boundaries did not significantly change with the changing threshold for a superpixel to be in the neighborhood region of another superpixel. As a result of this situation the required number of superpixels is fixed to 1000 and the experiment is repeated by introducing a multiplier that will make the radius larger for it to cover more superpixels. The results are shown in Figure 23.

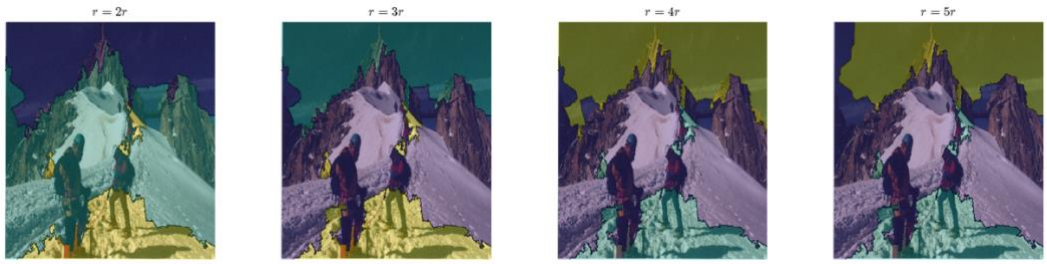

Fig. 23 Clusters generated using different radius multipliers.

From the results of changing the radius multiplier, the cluster boundaries did not significantly change. This might be occurred because of the similar neighbor members for the pixels close to each other. Therefore, the results stayed much or less the same.

The final results for the Part 4 of this homework are obtained by the parameter set defined in the Table 1 and the corresponding results can be seen from Figure 24.

Parameter	Value
Number of superpixels	200
Compactness parameter	10
Wavelengths	[2 4 8 16]
Orientations	[0 30 60 90]
Pixel Threshold	1000
Radius Multiplier	2

Table 1 The set of parameters for the final set-up

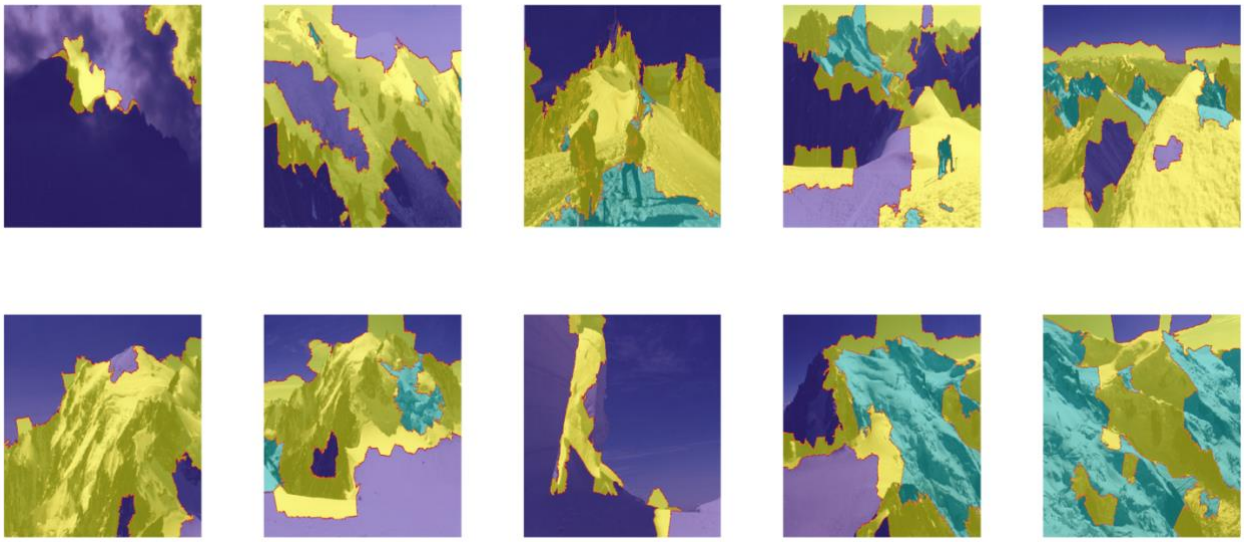

Fig. 24 Final results for applying segmentation using extended filters with the described algorithm.

From the results it can be stated that extending the feature vectors using the mean features of the neighbor superpixels increased the performance of the segmentation algorithm slightly but not preferably by considering the required computational power.

Conclusion

Aim of this assignment is to understand and implement the classical techniques for image segmentation. Concepts like SLIC, Gabor filtering, k-means clustering are explored using a challenging set of data. The optimization of the parameters of the algorithms by making informed judgments was one of the important outcomes of this task. Overall, like the previous homework this one was also very informative about the hands-on concepts of computer vision.

References

[1] R. Achanta, A. Shaji and K. Smith, "SLIC Superpixels Compared to State-of-the-Art Superpixel Methods," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 34, no. 11, pp. 2274-2281, 2012.

[2] IVRL, "SLIC Superpixels," EPFL, [Online]. Available: https://www.epfl.ch/labs/ivrl/research/slic-superpixels/. [Accessed 12 May 2024].

[3] MathWorks, "gabor," MATLAB, [Online]. Available: https://www.mathworks.com/help/images/ref/gabor.html. [Accessed 12 May 2024].

[4] MathWorks, "imgaborfilt," MATLAB, [Online]. Available: https://www.mathworks.com/help/images/ref/imgaborfilt.html. [Accessed 12 May 2024].

[5] MathWorks, "edge," MATLAB, [Online]. Available: https://www.mathworks.com/help/images/ref/edge.html. [Accessed 12 May 2024].