Atahan İyiekici
21801985
CS484-001
HW3-Report

CS484 HW3 Report

External libraries that I used in the homework:

from skimage.segmentation import slic, mark_boundaries

from sklearn.cluster import KMeans

Results of Part 1:

Description of the parameters in Part 1:

- **n_segments:** The approximate number of superpixels in the picture is controlled by this parameter. Increasing n_segments parameter results in generating more superpixels with smaller sizes. Also, increasing n_segments increases the computational complexity.
- **compactness:** This parameter balances color and spatial closeness. Higher values give more weight to spatial closeness, resulting in more square/cubic superpixel forms. Lower values emphasize color closeness, making superpixel forms more reliant on picture content.

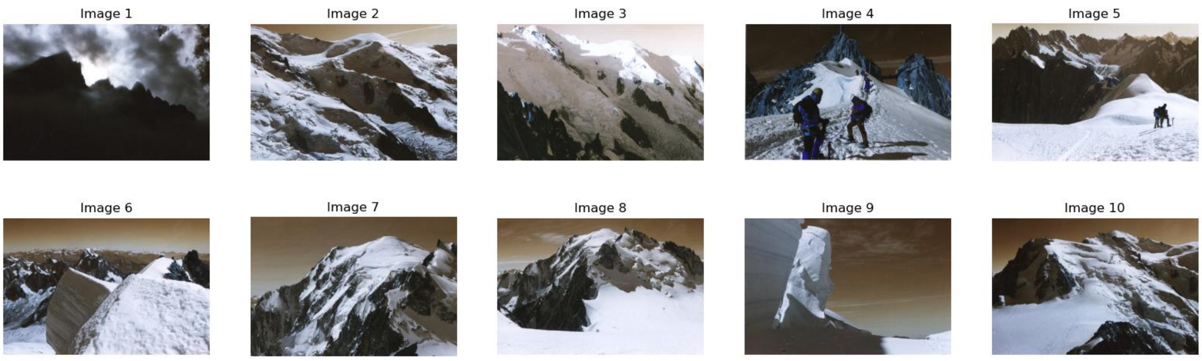

Figure 1: Original images

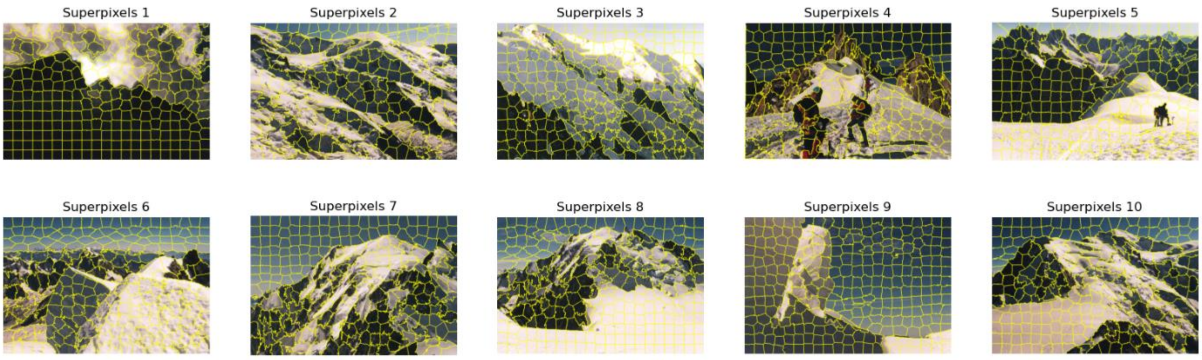

Figure 2: Superpixels of images with default parameters (n_segments = 300, compactness = 10)

Atahan İyiekici
21801985
CS484-001
HW3-Report

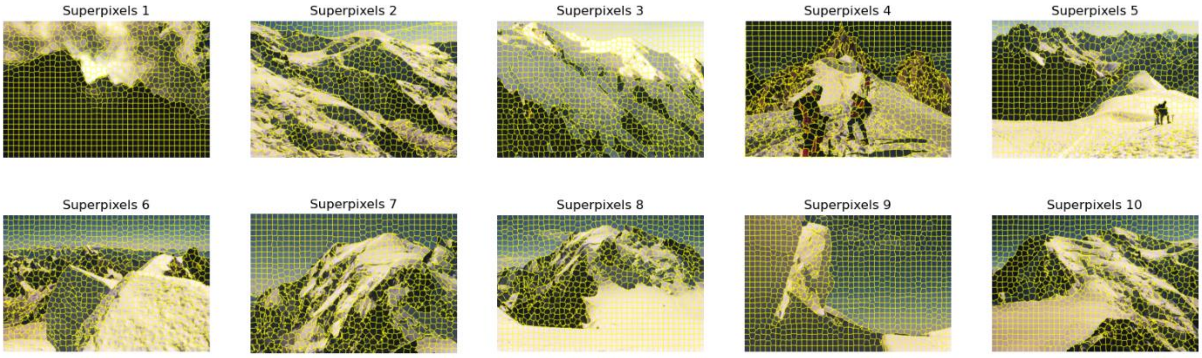

Figure 3: Superpixels of images with different parameters (n_segments = 1000, compactness = 10)

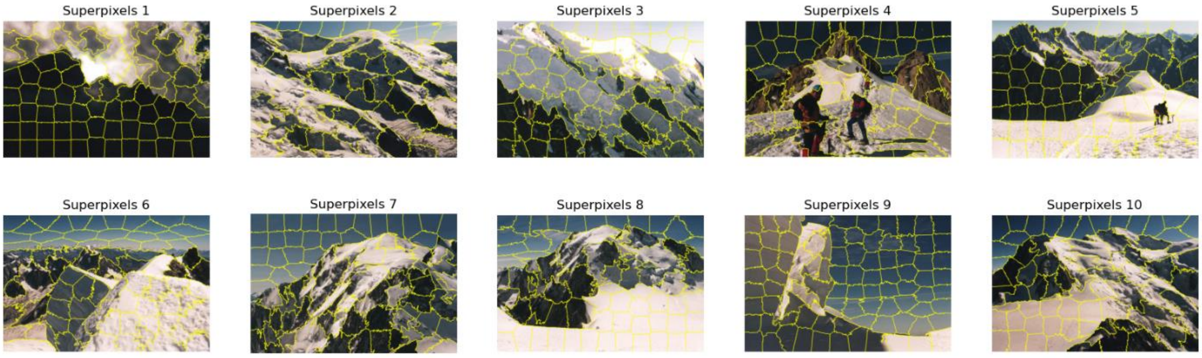

Figure 4: Superpixels of images with different parameters (n_segments = 100, compactness = 10)

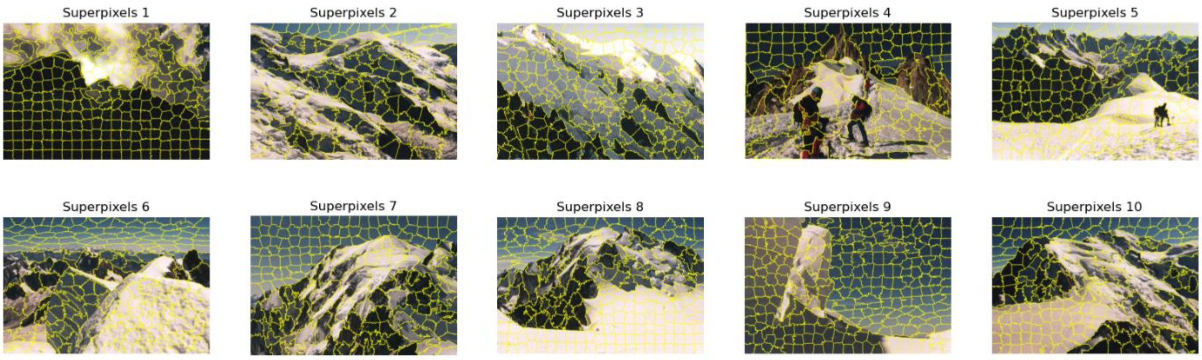

Figure 5: Superpixels of images with different parameters (n_segments = 300, compactness = 5)

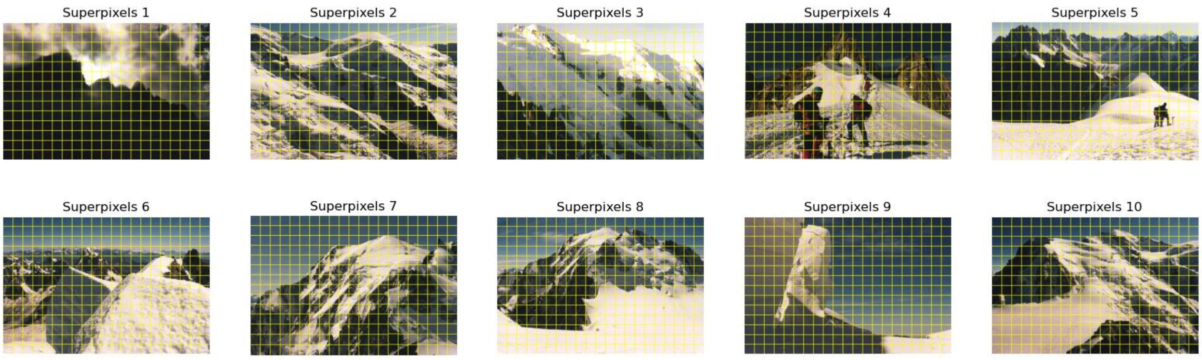

Figure 6: Superpixels of images with different parameters (n_segments = 300, compactness = 500)

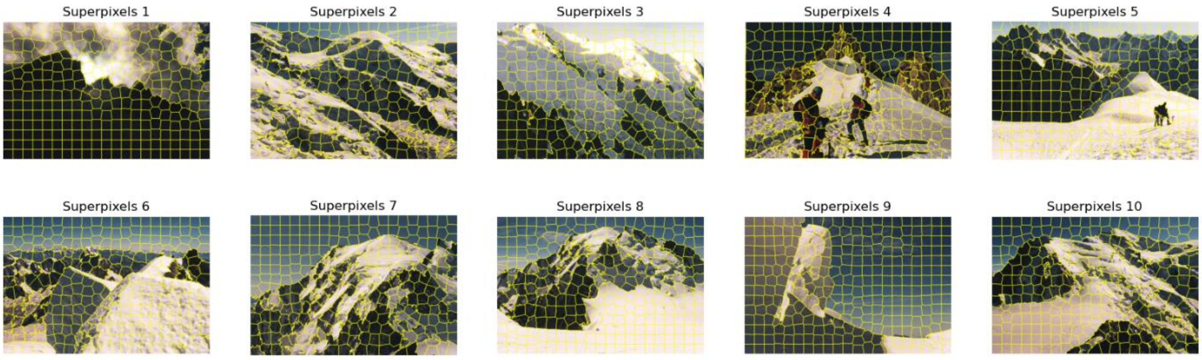

Figure 7: Superpixels of images with chosen parameters (n_segments = 300, compactness = 25)

Results of Part 2:

Description of the extra parameters in Part 2:

- **num_scales:** The number of scales (i.e., sizes) of the Gabor filters is determined by this parameter. A greater value will collect texture information at a wider range of sizes.
- **num_orientations:** The number of orientations of the Gabor filters is determined by this parameter. A greater number will collect texture information at a wider range of angles.

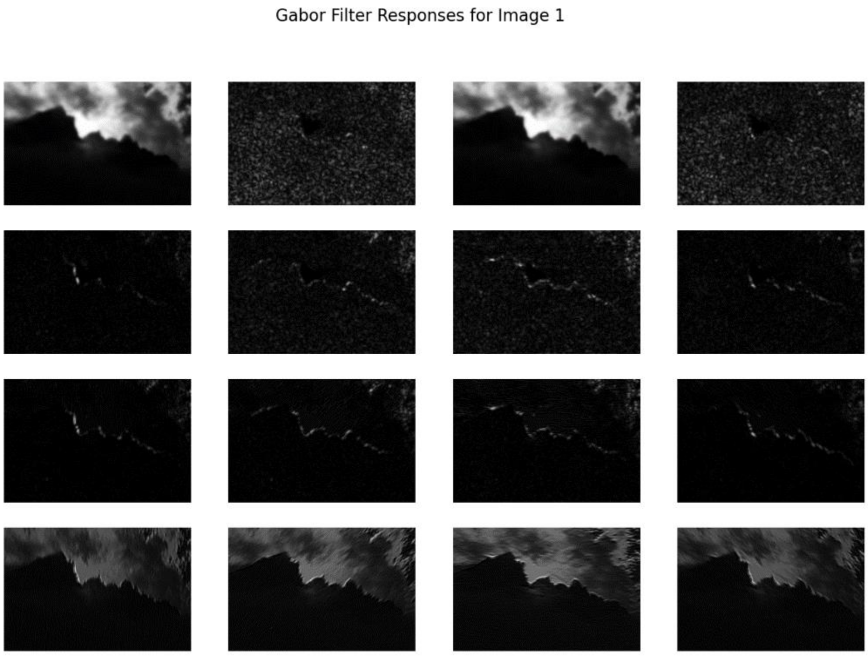

Figure 8: Gabor filter features for Image 1 where (orientation = 4, scale = 4)

Atahan İyiekici
21801985
CS484-001
HW3-Report

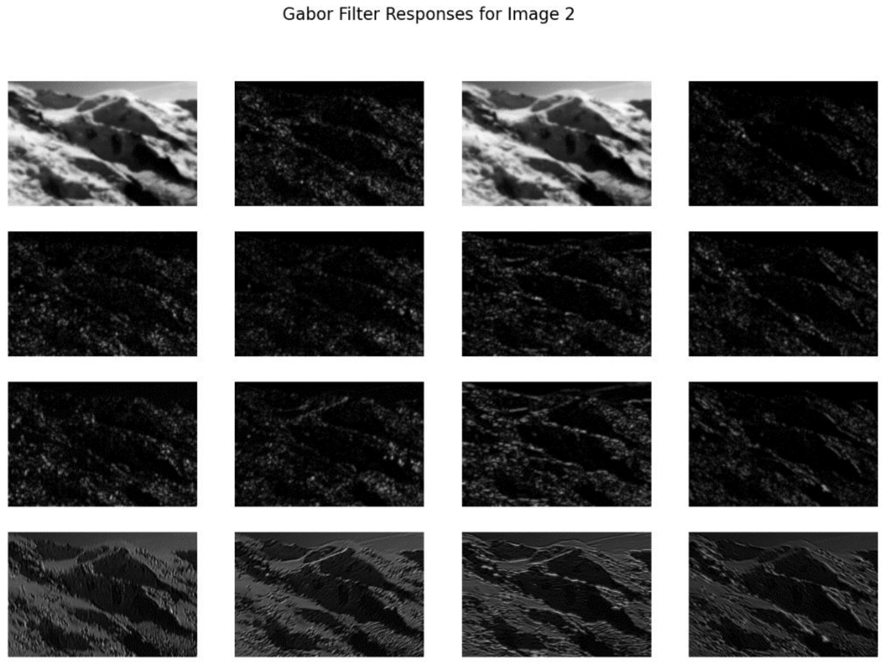

Figure 9: Gabor filter features for Image 2 where (orientation = 4, scale = 4)

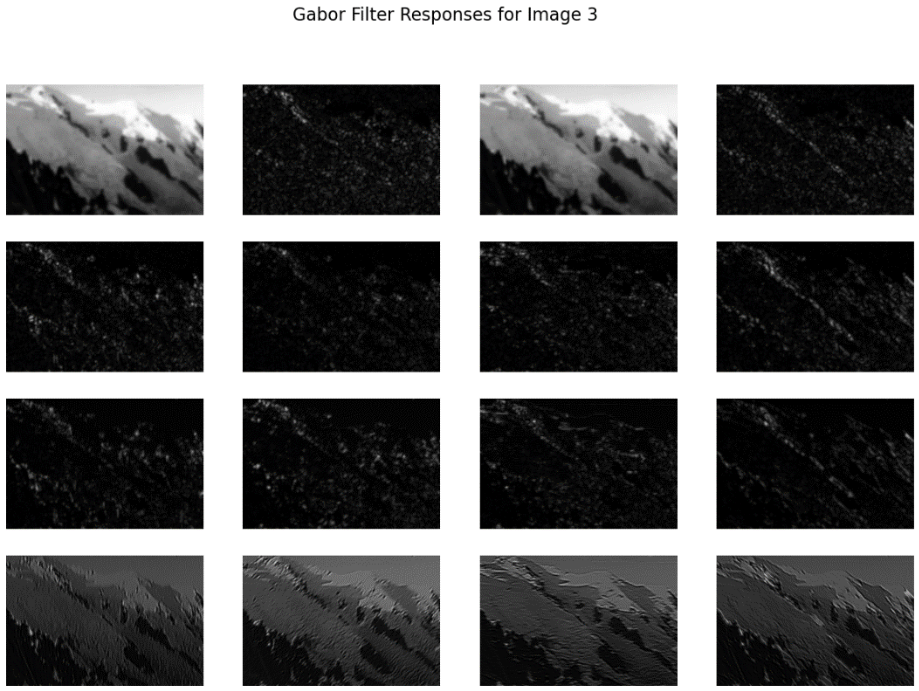

Figure 10: Gabor filter features for Image 3 where (orientation = 4, scale = 4)

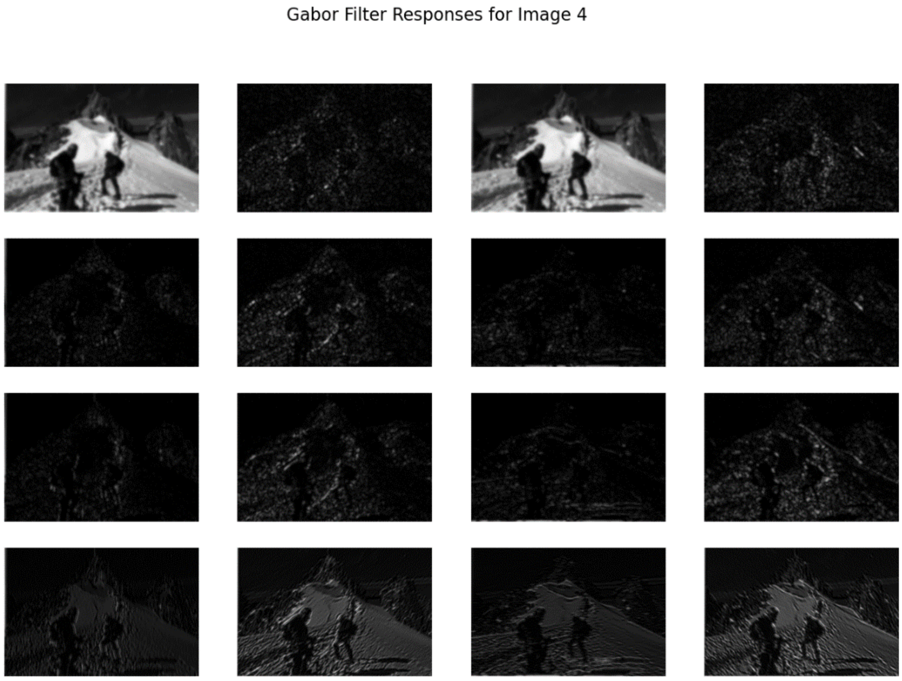

Figure 11: Gabor filter features for Image 4 where (orientation = 4, scale = 4)

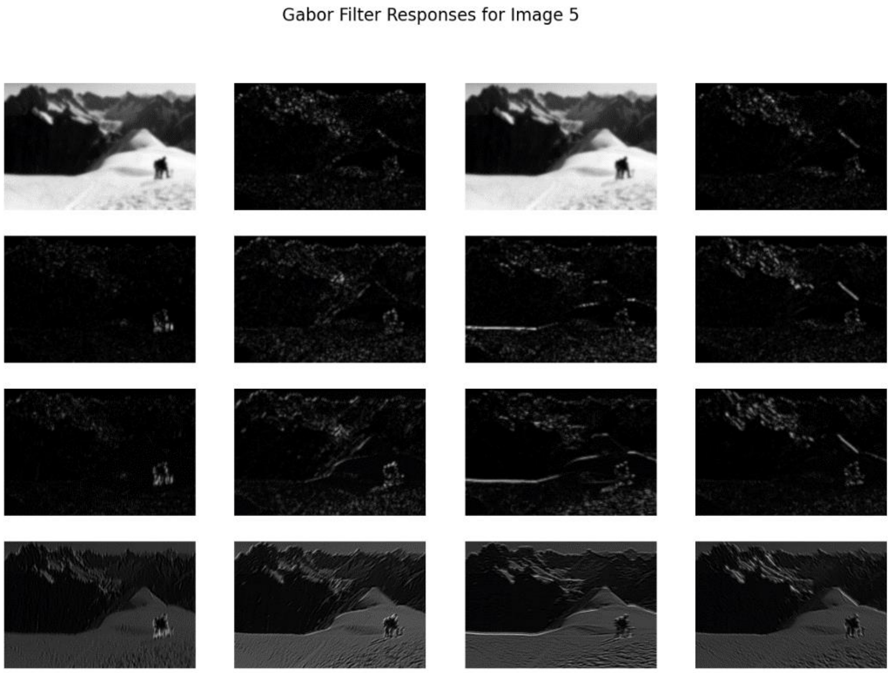

Figure 12: Gabor filter features for Image 5 where (orientation = 4, scale = 4)

Gabor Filter Responses for Image 6

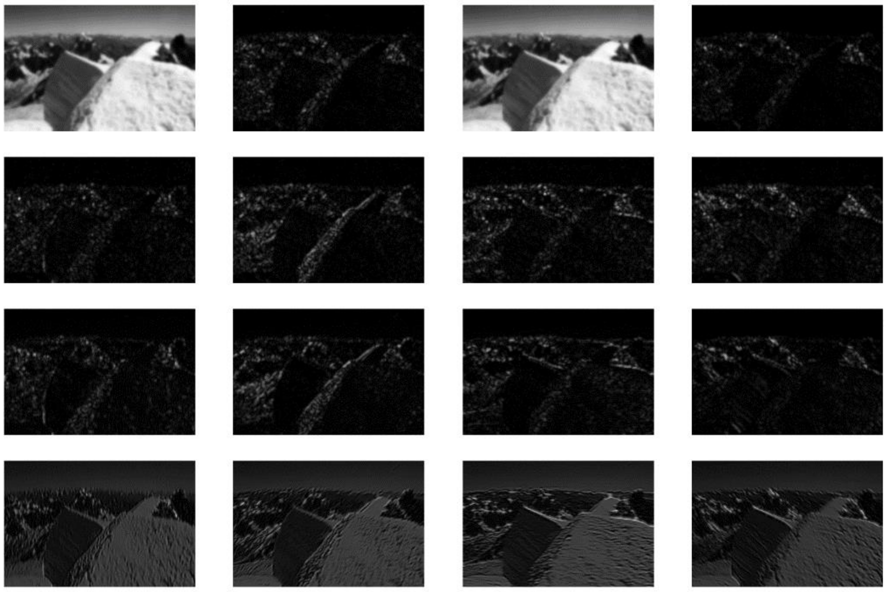

Figure 13: Gabor filter features for Image 6 where (orientation = 4, scale = 4)

Gabor Filter Responses for Image 7

Figure 14: Gabor filter features for Image 7 where (orientation = 4, scale = 4)

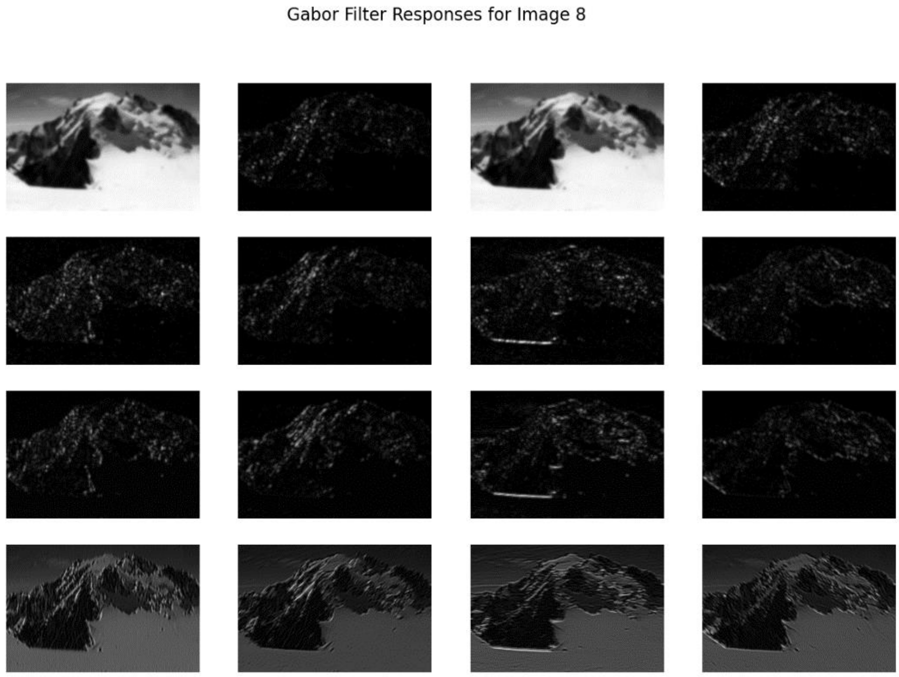

Figure 15: Gabor filter features for Image 8 where (orientation = 4, scale = 4)

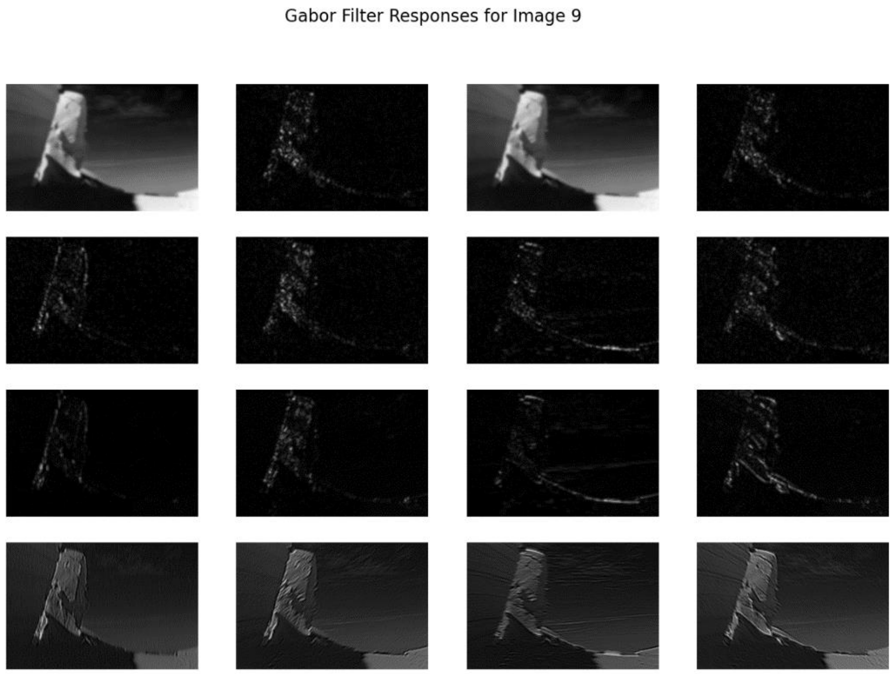

Figure 16: Gabor filter features for Image 9 where (orientation = 4, scale = 4)

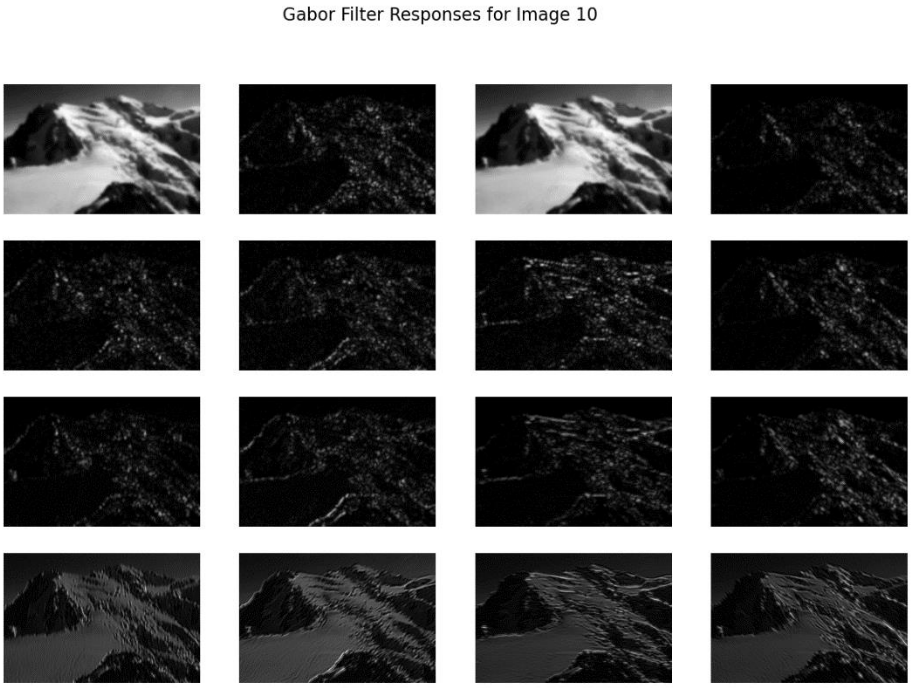

Figure 17: Gabor filter features for Image 10 where (orientation = 4, scale = 4)

Results with different orientation and scale:

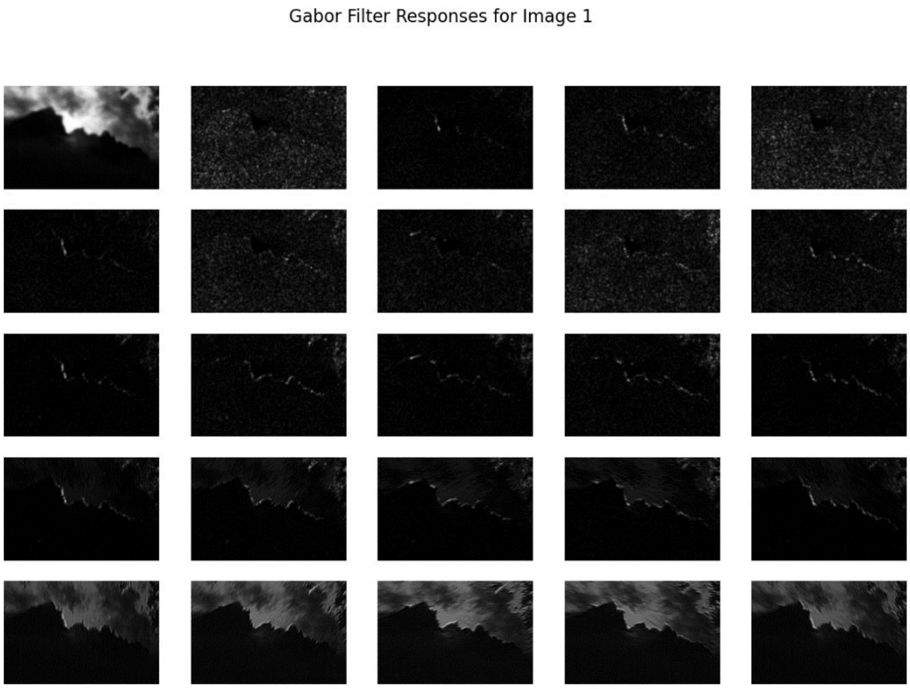

Figure 18: Gabor filter features for Image 1 where (orientation = 5, scale = 5)

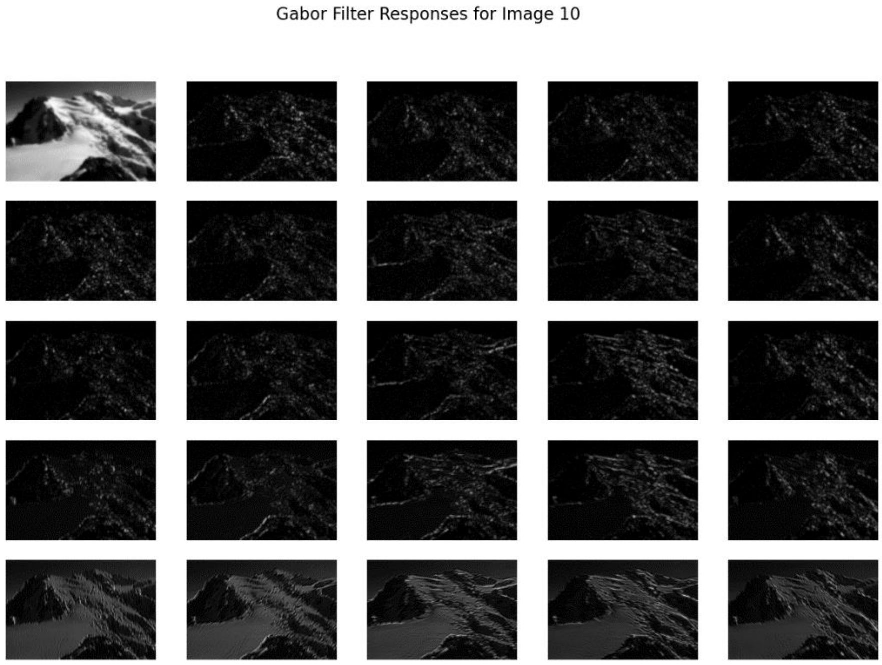

Figure 19: Gabor filter features for Image 10 where (orientation = 5, scale = 5)

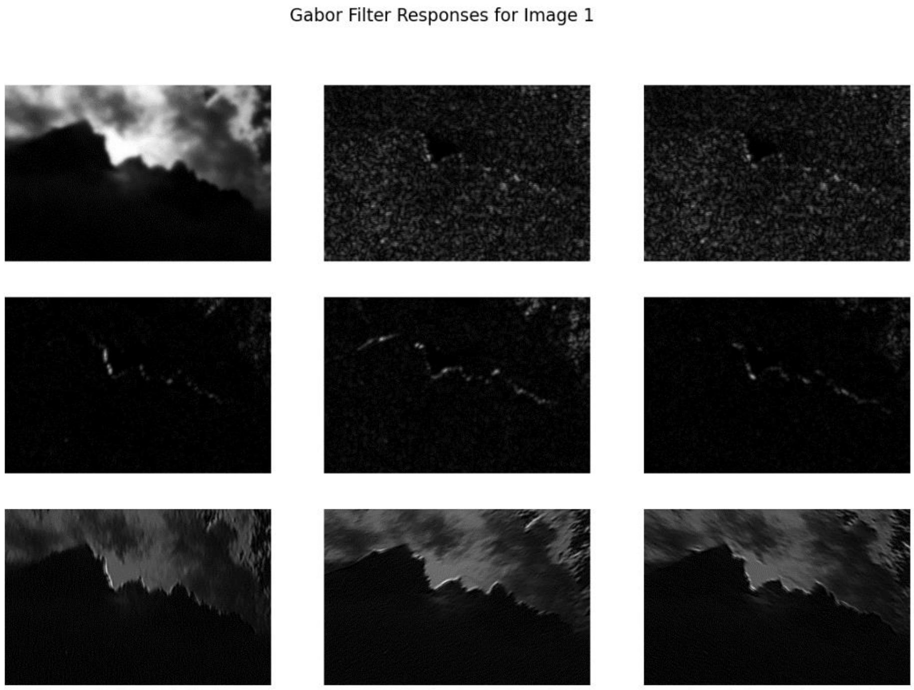

Figure 20: Gabor filter features for Image 1 where (orientation = 3, scale = 3)

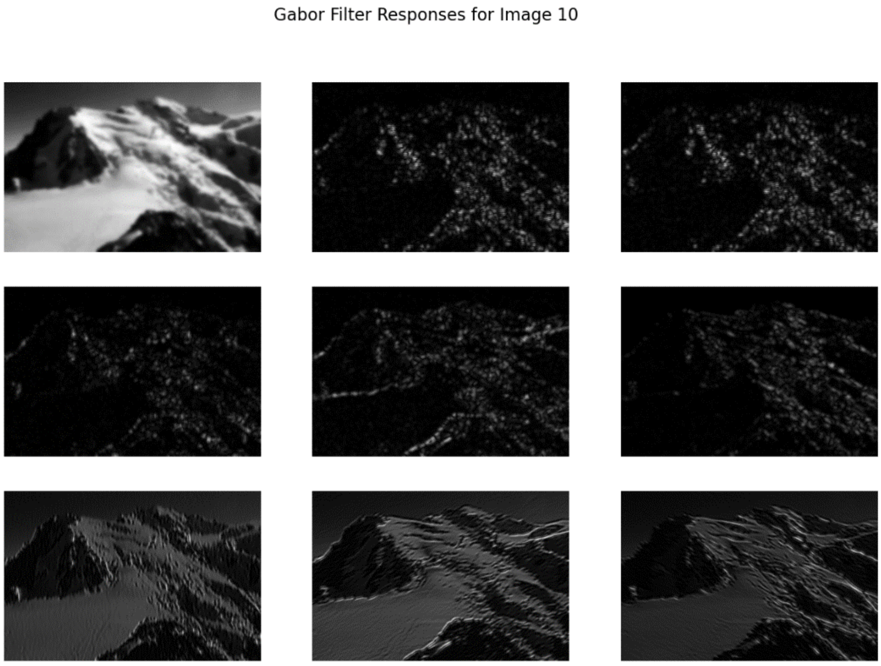

Figure 21: Gabor filter features for Image 10 where (orientation = 3, scale = 3)

Note: Other parameters of gabor filter stayed the same as their default values.

Results for Part 3:

Description of the extra parameters in Part 3:

- **num_clusters:** The number of clusters created by the k-means clustering method is determined by this parameter. A higher score indicates a greater number of smaller clusters.

Figure 22: Segmentation results for different parameters

(n_segments = 300, compactness = 25, num_clusters = 4)

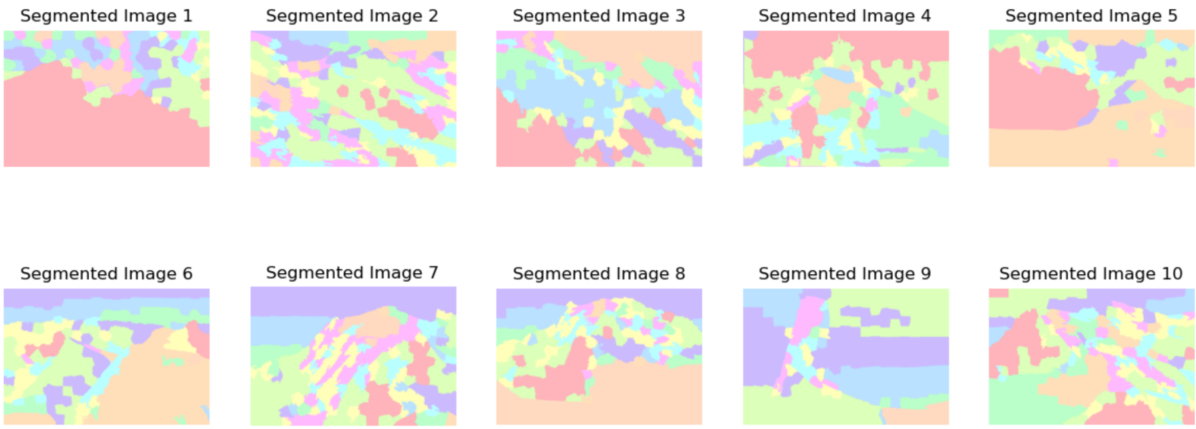

Figure 23: Segmentation results for different parameters

(n_segments = 300, compactness = 25, num_clusters = 10)

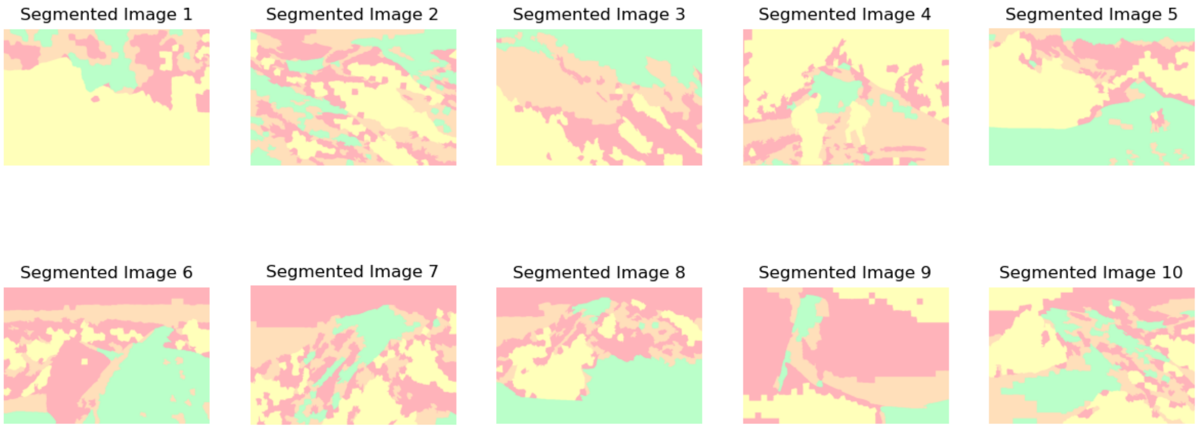

Figure 24: Segmentation results for different parameters

(n_segments = 800, compactness = 25, num_clusters = 4)

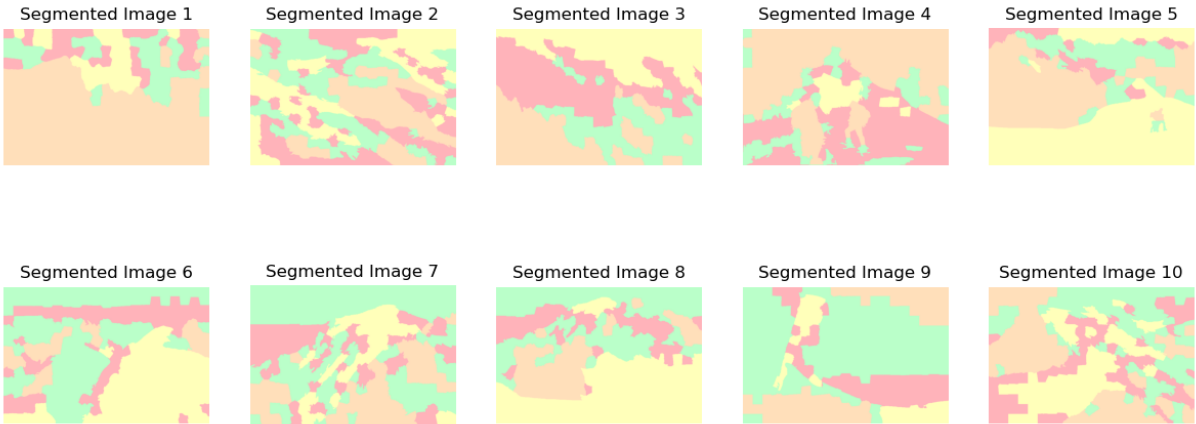

Figure 25: Segmentation results for different parameters

(n_segments = 300, compactness = 50, num_clusters = 4)

Figure 26: Segmentation results for different parameters

(n_segments = 300, compactness = 10, num_clusters = 4)

Figure 27: Segmentation results for different parameters

(n_segments = 300, compactness = 25, num_clusters = 3)

Results for Part 4:

Description of the extra parameters in Part 4:

- **pixel_threshold:** This option determines whether a superpixel is within a ring surrounding a certain superpixel. It is the bare minimum of pixels necessary to accept a superpixel within a specific ring.

Figure 28: Contextual Segmentation results for different parameters

(n_segments = 300, compactness = 25, num_clusters = 4, pixel_threshold = 10)

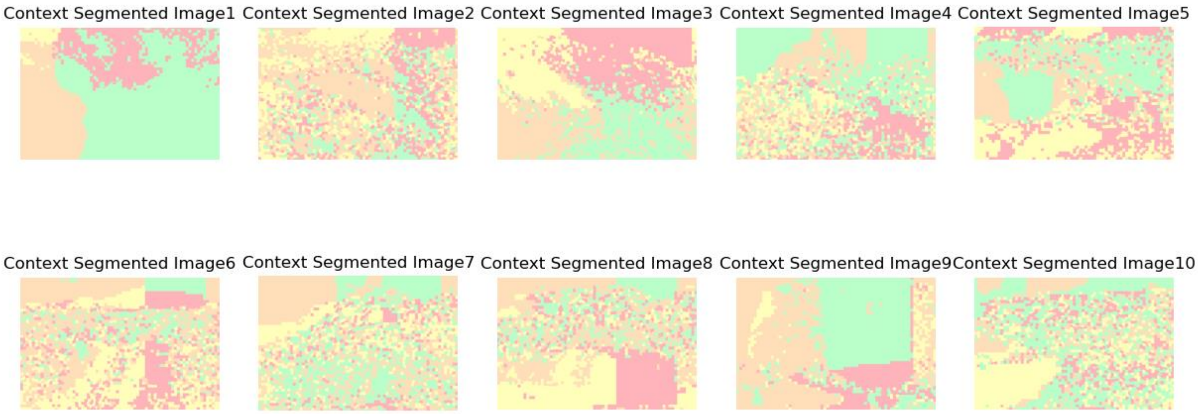

Figure 29: Contextual Segmentation results for different parameters

(n_segments = 3000, compactness = 25, num_clusters = 4, pixel_threshold = 10)

Figure 30: Contextual Segmentation results for different parameters

(n_segments = 300, compactness = 25, num_clusters = 4, pixel_threshold = 3)

Figure 31: Contextual Segmentation results for different parameters

(n_segments = 300, compactness = 25, num_clusters = 4, pixel_threshold = 0.5)

Figure 32: Contextual Segmentation results for different parameters

(n_segments = 300, compactness = 50, num_clusters = 4, pixel_threshold = 10)

Figure 33: Contextual Segmentation results for different parameters

(n_segments = 10, compactness = 25, num_clusters = 4, pixel_threshold = 10)

Figure 34: Contextual Segmentation results for different parameters

(n_segments = 15, compactness = 25, num_clusters = 4, pixel_threshold = 3)

Figure 35: Contextual Segmentation results for different parameters

(n_segments = 50, compactness = 23, num_clusters = 4, pixel_threshold = 5)

Figure 36: Contextual Segmentation results for different parameters

(n_segments = 50, compactness = 23, num_clusters = 4, pixel_threshold = 1)

Atahan İyiekici
21801985
CS484-001
HW3-Report

Detailed discussion of the results:

In the first part, I used slic algorithm to obtain superpixels for all images. There were 2 parameters which are n_segments and compactness. N_segments changes the superpixel number, and compactness balances color and spatial closeness. Firstly, I tried the default values as n_segments=300 and compactness=10, as shown in the Figure 2. Then, I increased and decreased the value of superpixel number and observe the change in the images in Figure 3 and Figure 4. In these cases I saw the when n_segments increases the superpixel number is increasing but the size of superpixel decreasing. On the other hand when I decrease the n_segments, superpixel number decreased and size of superpixels increased. Then, I stick with the value of 300 for n_segments, and change the values for compactness. In Figure 5, Figure 6 and Figure 7, I tried 3 different values for compactness. When I increase the compactness I saw that the superpixels becomes more square, vise versa, when I decrease the compactness the shapes of superpixels gets different than squares. I chose n_segments 300 and compactness 25 for this part for optimal values.

In the second part, for gabor filter responses and feature extractions, there were a lot of parameters but I used the default parameters for gabor filters. I, also, used kernel size 31, number of orientations as 4 and scale as 4. After displaying all images for these parameter values, I changed the orientation and scale to 3 and 5 to see what are these parameters effects on images. Then, I saw that increasing scale number and increasing orientation number gave me a lot different information about gabor filters used on images. Therefore, increasing these parameters gathers more texture information about the images. On the other hand, decreasing these values gave less information about the texture information of images. In my opinion, after increasing the values a lot, almost same information given repeatedly, therefore, increasing these values might be a waste of computational power. Also, choosing a small number for these values can lose information, therefore, the numbers should be chosen wisely.

In the third and forth parts, clustering and contextual representation with clustering parts, I used k means clustering algorithm in the scikit-learn library. For the part 3, there were 3 important parameters which are n_segments, compactness and number of clusters. Firstly, in the optimal result, in my opinion, I chose n_segments = 300, compactness = 25, number of clusters = 4, also the gabor filter segments are remained the same as the manual requested. As it might be seen, especially in figure 22 of images 1,4 and 5 the segmentation is well seperates the objects. For example, in image 4, two people are colored red and can understandable that they are human. Also, in the other figures the mountain boundaries and some parts are well segmented. To find the optimal results, I firstly stayed with the best parameters in the first part, and change the number of clusters. Less than 4 was insufficient and more than 4 distorts some already well seperated clusters. These can be seen in the figure 22 and 23. After these, I changed the superpixel number both in increase and decrease, which can be seen in the figures 22 and 24, it changes the images a lot. Some of them are better with more segments but most of them were better in the Figure 22, in my opinion. Then, I also changed the compactness in figures 25 and 26. The impact of compactness, I guess, is more effective increasing compactness wipes so many things and lead some misinterpretations. Lastly for the part 3, in Figure 27, I wanted to show the optimal values with 3 clusters, which gave a decent result but 4 clusters gaves more accurate information about the images. After these, I moved on to part 4, which includes another parameter of pixel threshold, the option determines the ring neighbors the pixels around it. Therefore, it was quite impactful for the results. In my optimal values in the previous parts, n_segments =300, compactness = 25, number of clusters = 4 and a value for pixel threshold which I chose 10 at first, I get worse results compared to part 3. In the segmentation of part 3 gave me more accurate insights about images. Therefore, I change all parameters one by one to find a better result. Changing superpixel number has a huge effect on the segmentation, I firstly increment it enormously as changing it from 300 to 3000, which gave me no better result. Then, I change it to lower values, lower than 50, after that I got better results. But before that I changed compactness, number of clusters and pixel thresholds a lot of times to get better results. These attempts gave me slightly better results but not I expected. After all experiments I get the most similar result in the parameters of n_segments = 50, compactness = 23, num_clusters = 4, pixel_threshold = 1 in the figure 36.