CS 484: Introduction to Computer Vision

Spring 2024

Homework 3

Modules/Libraries Used

MATLAB

SLICO MATLAB code from the EPFL website[1] is used to calculate superpixels. **rgb2gray** function is used for converting the images to grayscale. **imgaborfilt** function is used to filter images.

Python

loadmat function from the **SciPy** library is used to load the MATLAB .mat file containing superpixel labels and Gabor filter magnitudes. The **Pathlib** module is used for loading images. **Numpy** is used for mathematical operations. **Scikit Learn** is used for K-Means clustering. **Matplotlib** is used for reading, displaying images and plotting.

The Data

The dataset consists of 10 RGB images of rocky and snowy mountains. All images have a resolution of 756x502 (except for images 4 and 6 which have resolutions of 756x504 and 756x512).

[1] https://www.epfl.ch/labs/ivrl/wp-content/uploads/2018/08/SLIC_mex.zip

Figure 1: Image 1

Figure 2: Image 2

Figure 3: Image 3

Figure 4: Image 4

Figure 5: Image 5

Figure 6: Image 6

Figure 7: Image 7

Figure 8: Image 8

Figure 9: Image 9

Figure 10: Image 10

Part 1 (Superpixels)

The **SLICO** algorithm is used for superpixel oversegmentation. The number of required superpixels, which is the only parameter of the SLICO algorithm, is set to **450**. This value is used because it produces both a reasonable and identical number of superpixels in all images. The images after the superpixel oversegmentation process can be seen below.

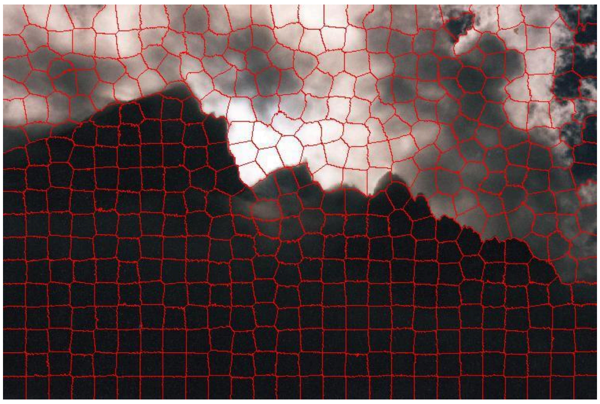

Figure 11: Image 1 with red superpixel borders

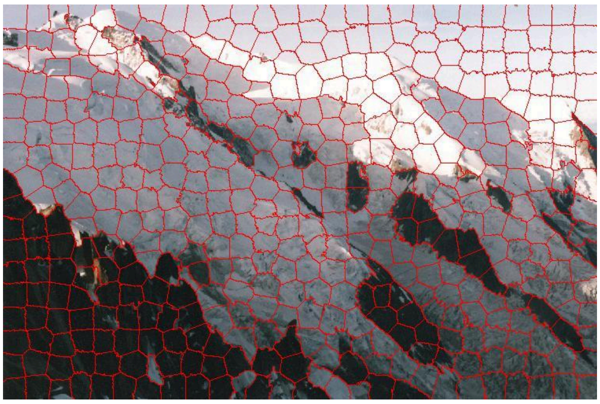

Figure 12: Image 2 with red superpixel borders

Figure 13: Image 3 with red superpixel borders

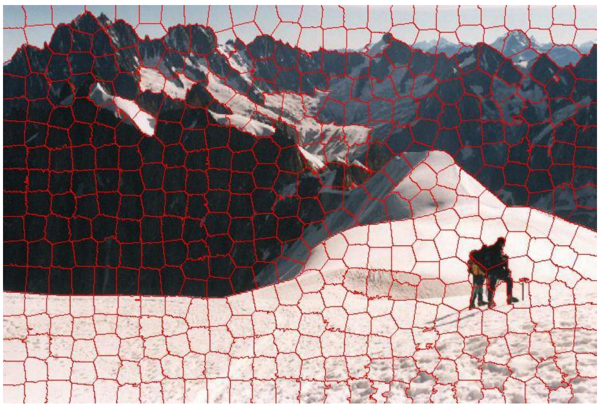

Figure 14: Image 4 with red superpixel borders

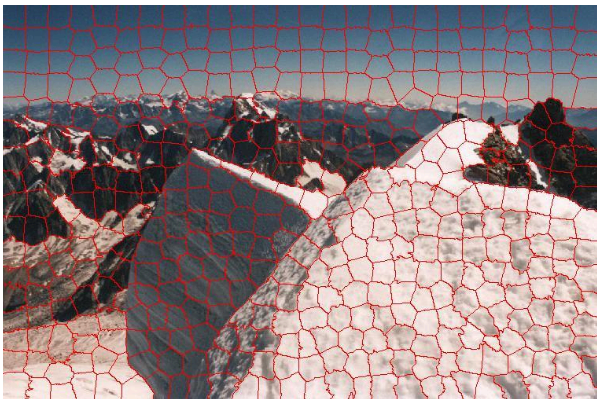

Figure 15: Image 5 with red superpixel borders

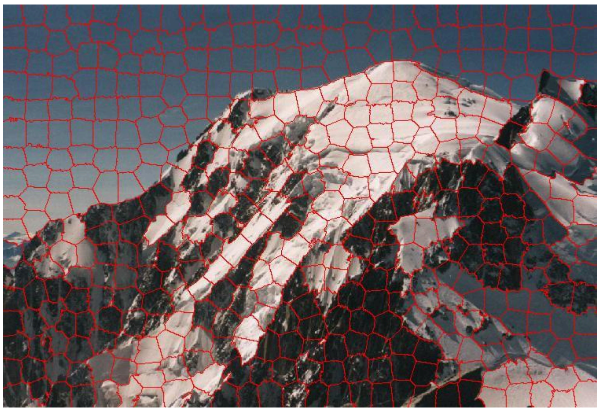

Figure 16: Image 6 with red superpixel borders

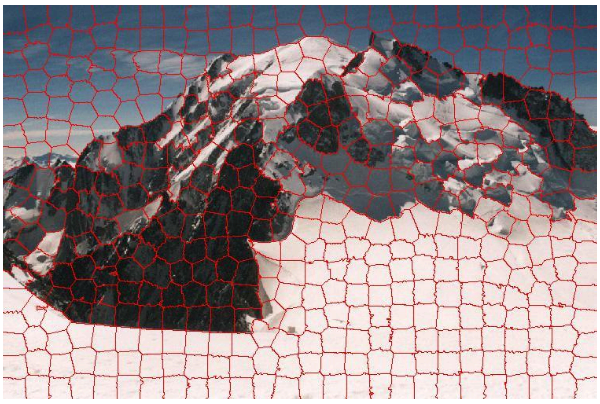

Figure 17: Image 7 with red superpixel borders

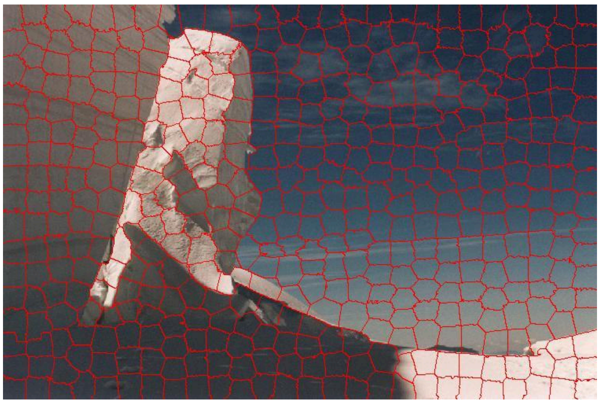

Figure 18: Image 8 with red superpixel borders

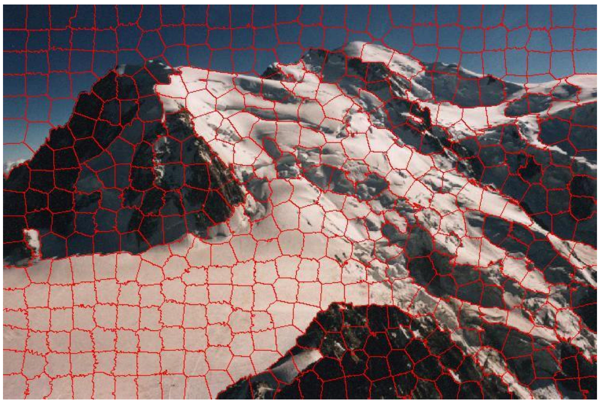

Figure 19: Image 9 with red superpixel borders

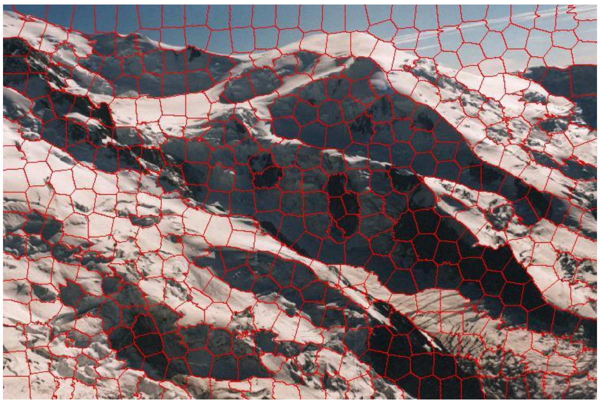

Figure 20: Image 10 with red superpixel borders

Part 2 (Gabor Filtering)

A Gabor filter bank with **5** wavelengths and **5** orientations is used. The wavelengths of the filters are **2, 4, 8, 16, 32** and the orientations of the filters are **0°, 36°, 72°, 108°, 144°**. These values are chosen to capture a wide variety of features from the image. Some examples of the outputs of the filters are as follows.

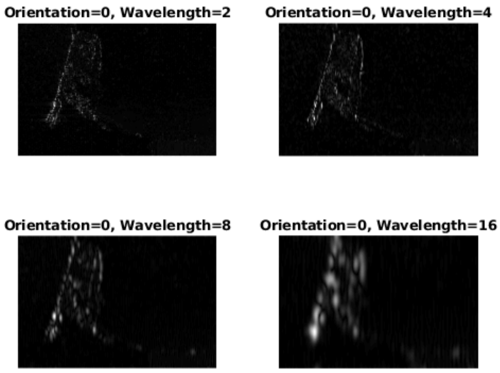

Figure 21: Same orientation, varying wavelengths (image 8)

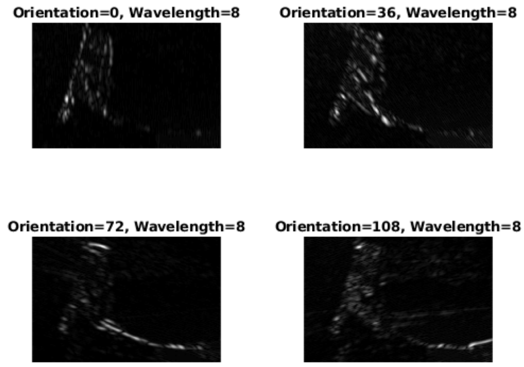

Figure 22: Varying orientation, same wavelengths (image 8)

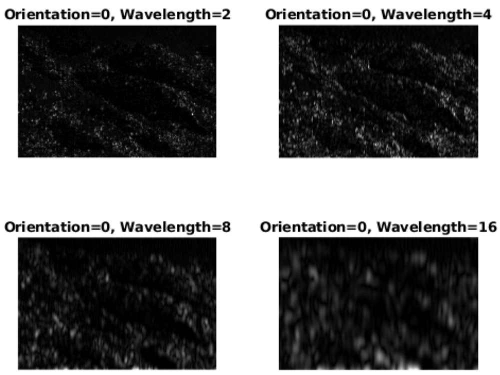

Figure 23: Same orientation, varying wavelengths (image 10)

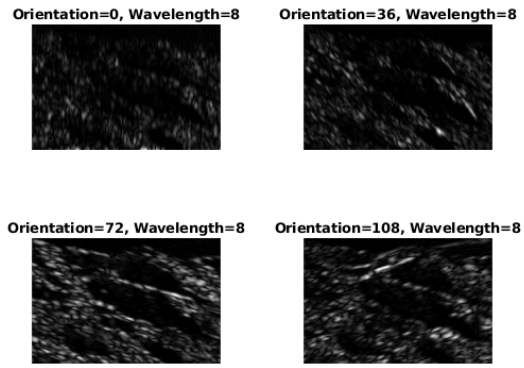

Figure 24: Varying orientation, same wavelengths (image 10)

Part 3 (Clustering)

After some experimentation with different algorithms such as mean shift, HDBSCAN, and spectral clustering, the **K-Means** algorithm is decided to give the best results. After some trial and error, the **K** value is chosen to be **3**. K values larger than 3 (and especially 4) result in many meaningless clusters. The images with clustered superpixels are as follows.

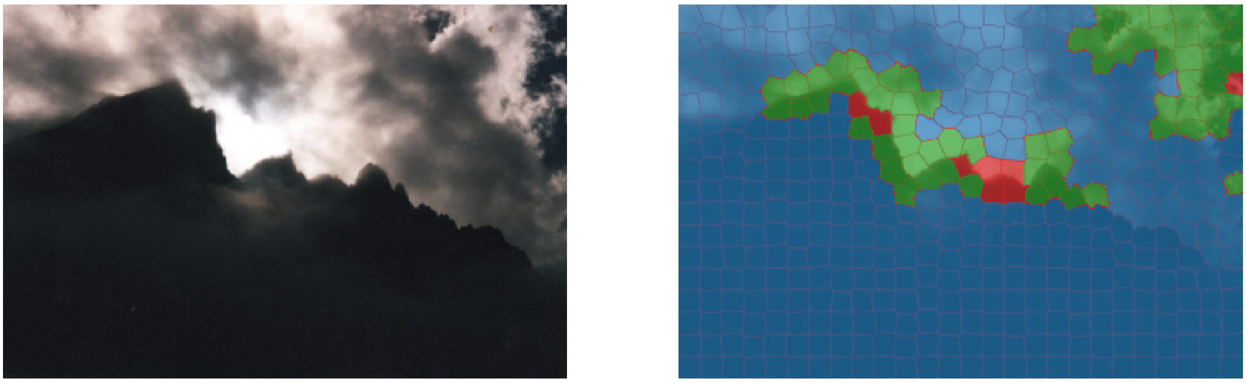

Figure 25: Image 1 with superpixels clustered by 25 Gabor filter features

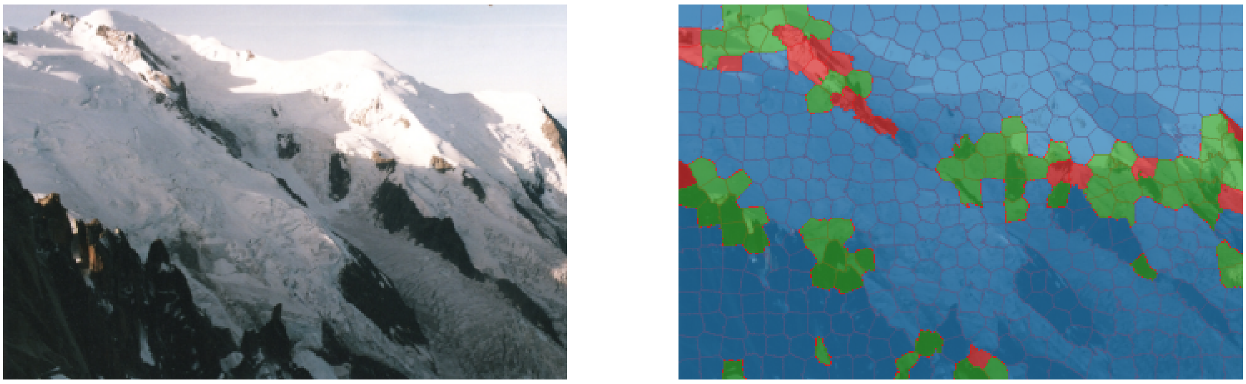

Figure 26: Image 2 with superpixels clustered by 25 Gabor filter features

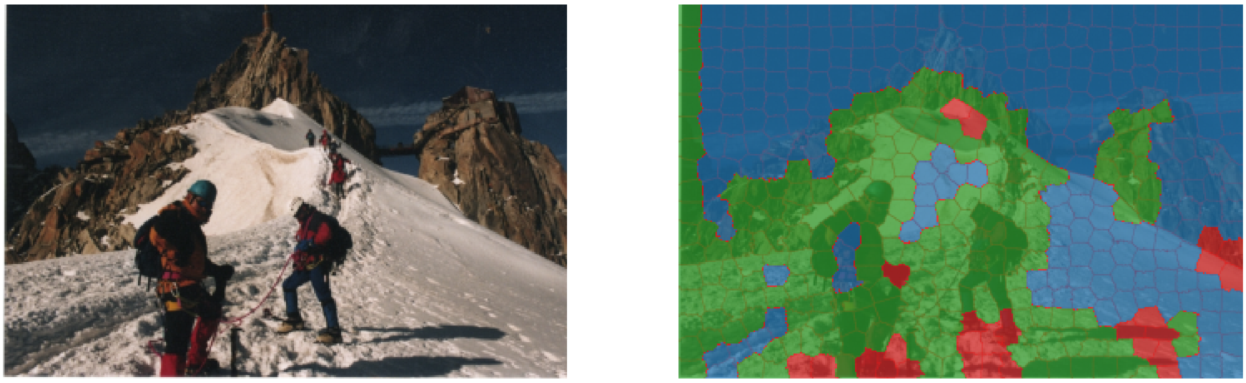

Figure 27: Image 3 with superpixels clustered by 25 Gabor filter features

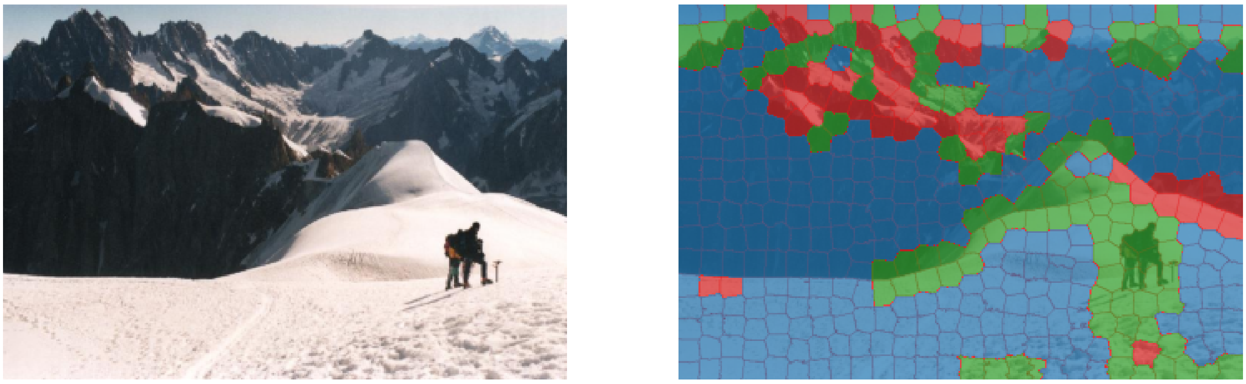

Figure 28: Image 4 with superpixels clustered by 25 Gabor filter features

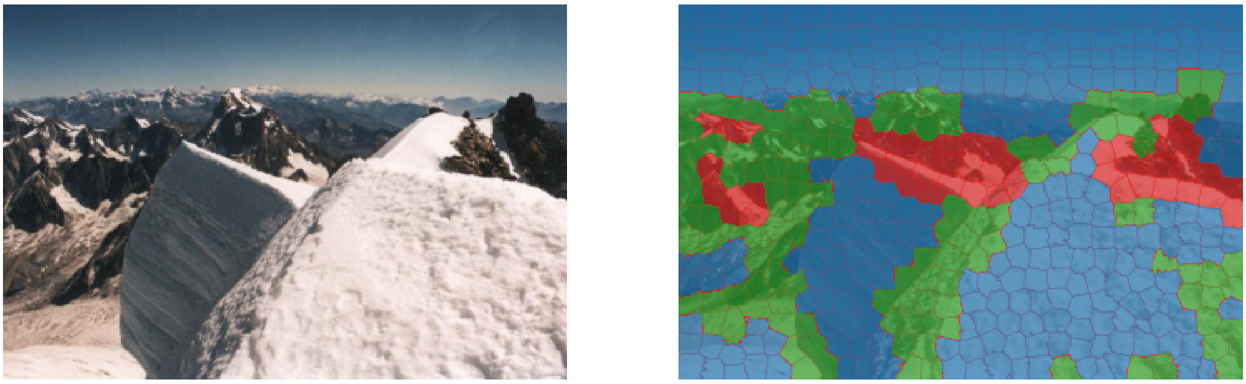

Figure 29: Image 5 with superpixels clustered by 25 Gabor filter features

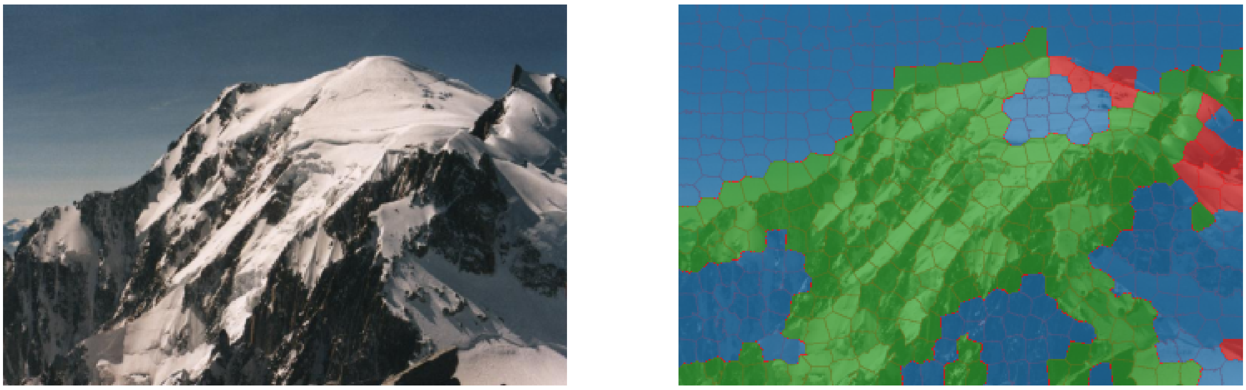

Figure 30: Image 6 with superpixels clustered by 25 Gabor filter features

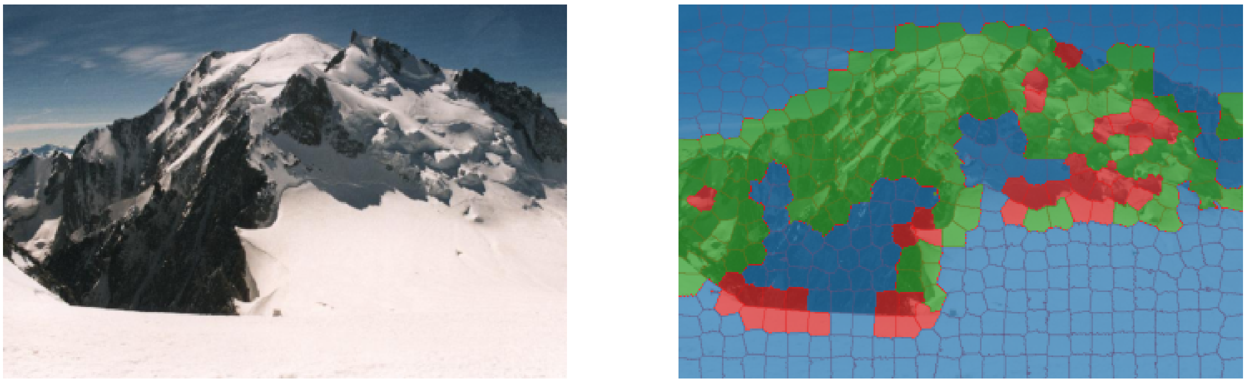

Figure 31: Image 7 with superpixels clustered by 25 Gabor filter features

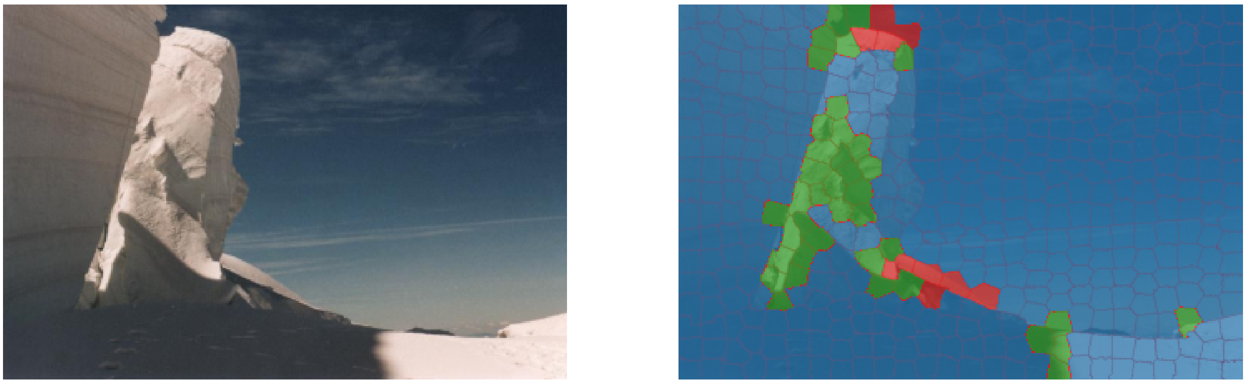

Figure 32: Image 8 with superpixels clustered by 25 Gabor filter features

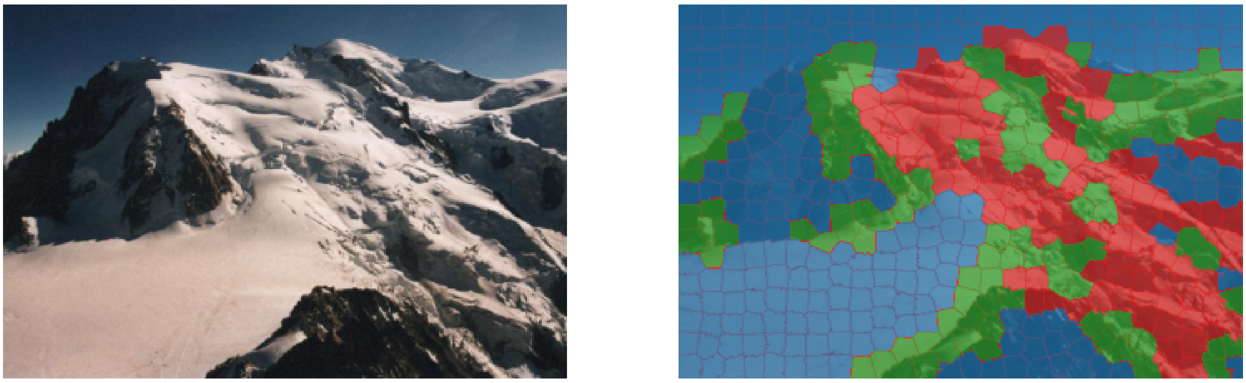

Figure 33: Image 9 with superpixels clustered by 25 Gabor filter features

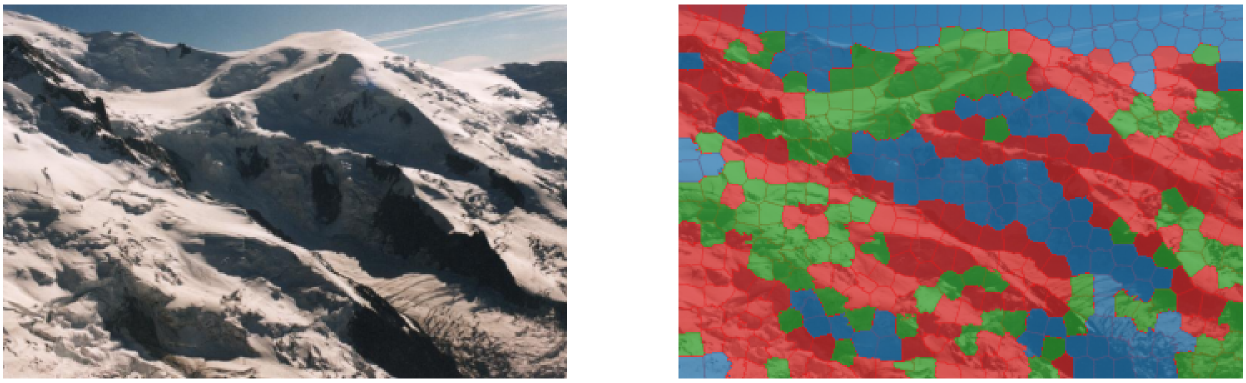

Figure 34: Image 10 with superpixels clustered by 25 Gabor filter features

Part 4 (Clustering With More Features)

After the calculation of the **centroids** of every superpixel, the **level 1 neighbors** of a superpixel is defined as other superpixels whose centroids are in the circle that is centered at the main superpixel and has a radius of 1.5x the diameter of the smallest circle that the main superpixel can fit inside. The **level 2 neighbors** of a superpixel is defined in a similar way with the radius multiplier being 2.5x rather than 1.5x and excluding the level 1 neighbors. The following images are the result of 3-Means clustering with 75 features, 25 features from before, 25 features calculated as the average of the feature values of the superpixels in the level 1 neighborhood, and 25 features calculated as the average of the feature values of the superpixels in the level 2 neighborhood.

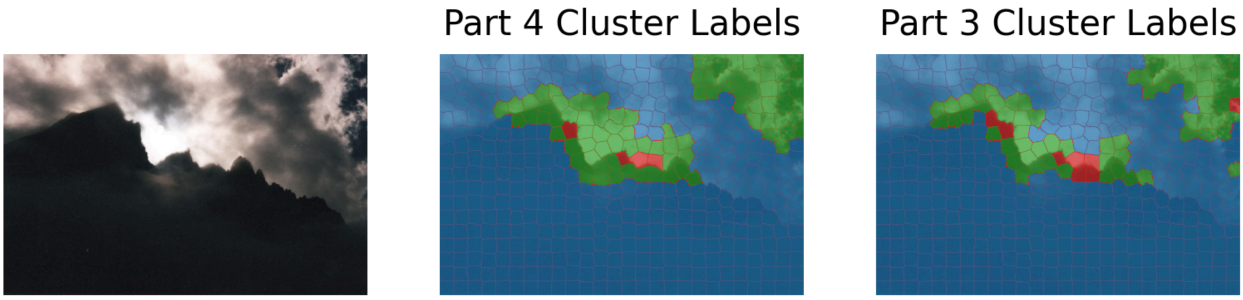

Figure 35: Image 1 with superpixels clustered by 75 Gabor filter features

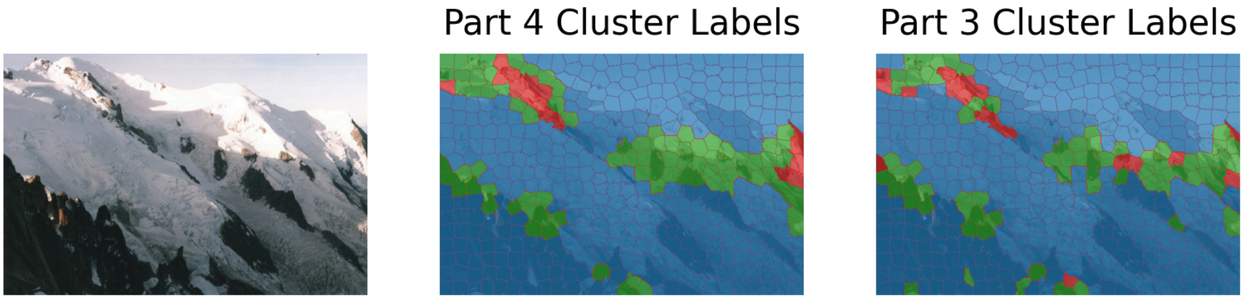

Figure 36: Image 2 with superpixels clustered by 75 Gabor filter features

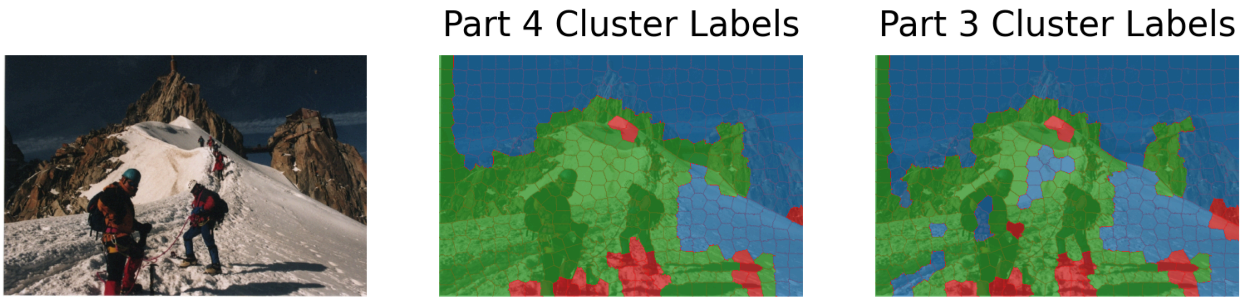

Figure 37: Image 3 with superpixels clustered by 75 Gabor filter features

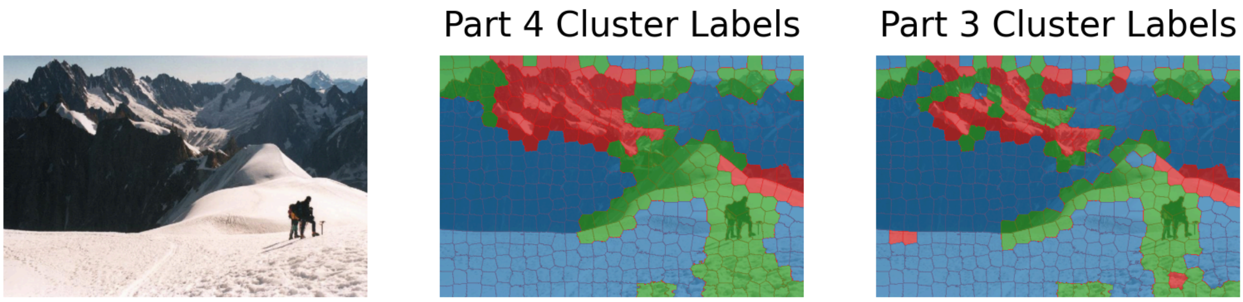

Figure 38: Image 4 with superpixels clustered by 75 Gabor filter features

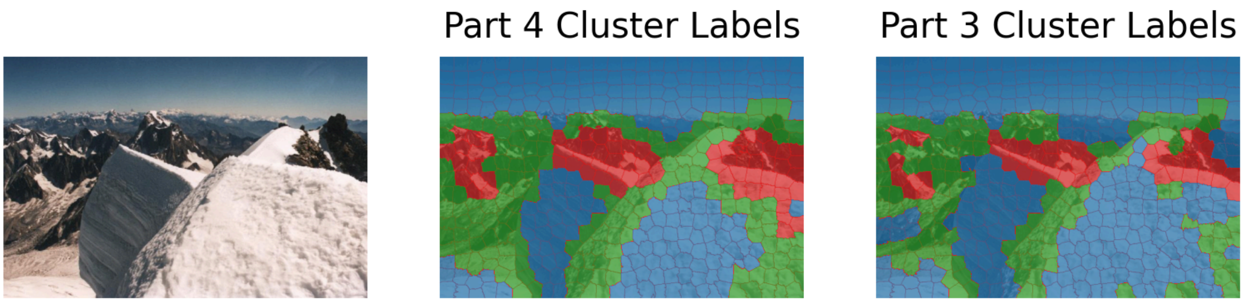

Figure 39: Image 5 with superpixels clustered by 75 Gabor filter features

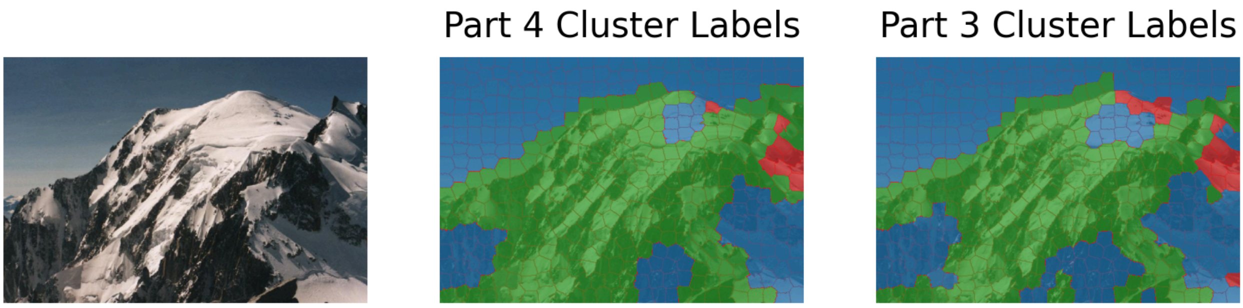

Figure 40: Image 6 with superpixels clustered by 75 Gabor filter features

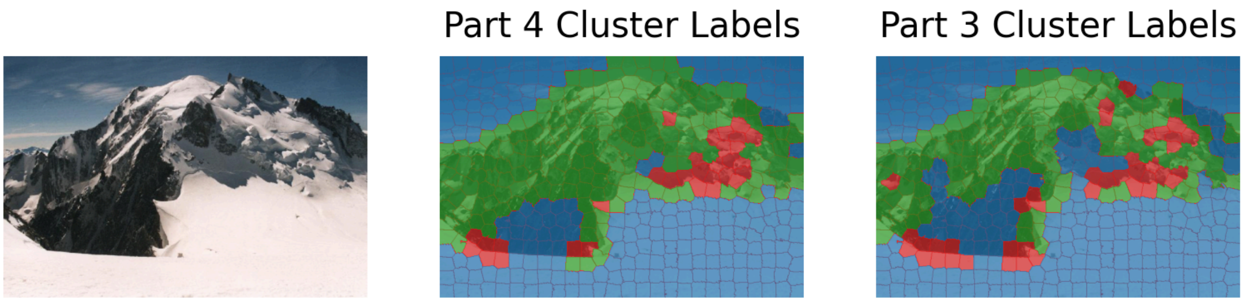

Figure 41: Image 7 with superpixels clustered by 75 Gabor filter features

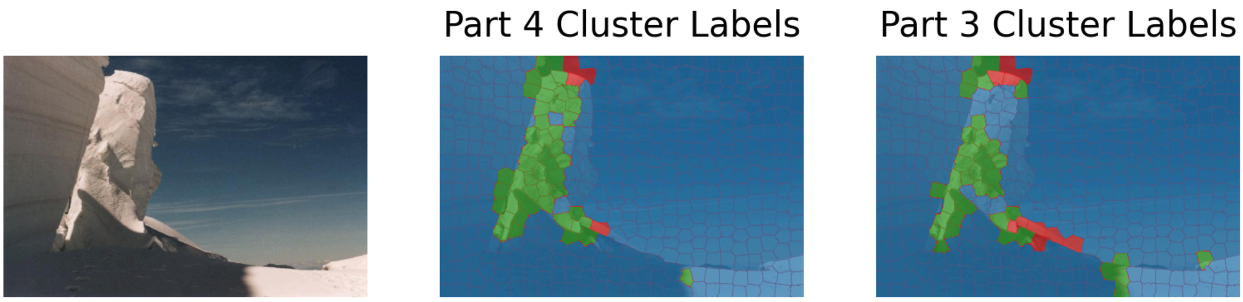

Figure 42: Image 8 with superpixels clustered by 75 Gabor filter features

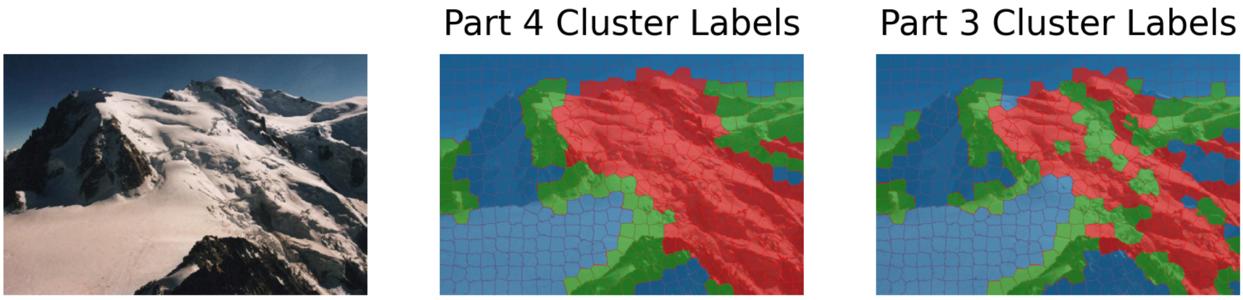

Figure 43: Image 9 with superpixels clustered by 75 Gabor filter features

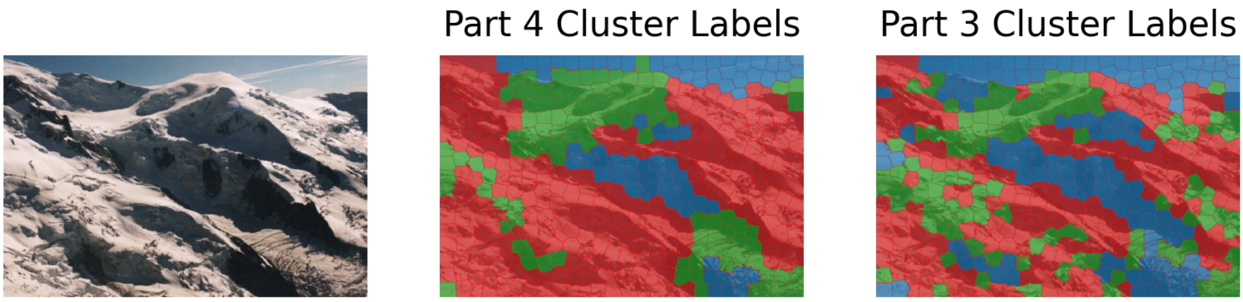

Figure 44: Image 10 with superpixels clustered by 75 Gabor filter features

Conclusion

The “number of required superpixels” value of the SLICO algorithm roughly determines the number of superpixels in the resulting image. Different images can have different numbers of superpixels even though the parameter value is the same. The number of superpixels is chosen to be appropriate for all images in the dataset.

The wavelengths and orientations of the gabor filter bank were chosen to capture details of varying size and angle. The sizes are equidistant on the log scale and the orientations divide 180° into 5 equal parts.

The number of means is chosen to be 3 because 2 is overly simplistic and 4 produces too many clusters. Blue labels seem to be superpixels with little texture, green labels seem to be superpixels with right leaning surfaces, and green labels seem to be superpixels with left leaning surfaces.

When the context of each superpixel is taken into account (part 4), the segmentation results are less abrupt/smoother.

In general, homogeneous areas are segmented more effectively compared to areas with more intricate detail.