CV Assignment 3

Name: Meetakshi Setiya
Roll no: 2019253

Please be mindful of the directory structure. I have assumed that the output and input folders are present and created as is.

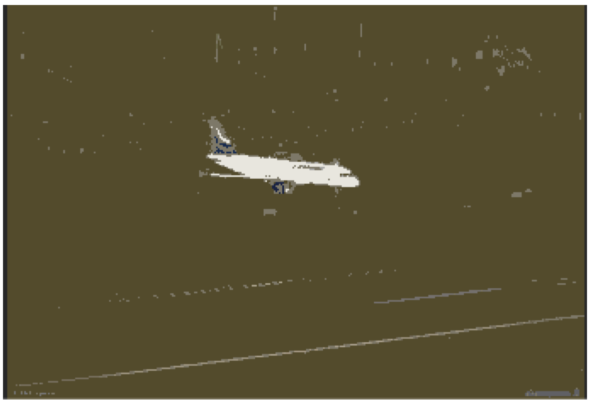
1.

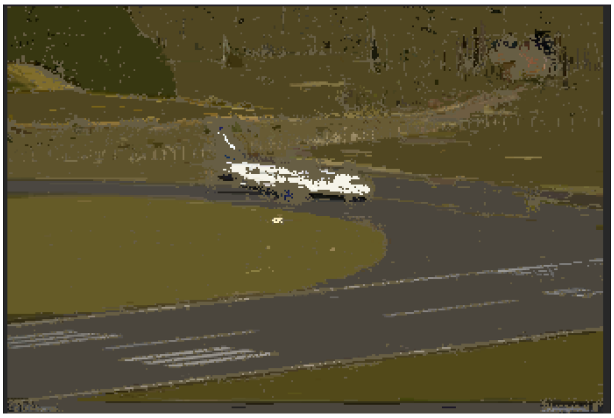
I have taken the number of super-pixels to be 150 since I found a good output at that point. The output in the outputs folder is for 150 superpixels. The program would however ask the user for inputting the path to input image, output directory and the number of superpixels. The output image is q1_saliency_map.png

2.

In DBSCAN, I have used some values of epsilon i.e. the search radius and minpts i.e. the minimum number of points required to deem a point as the core point.
Once DBSCAN returned clusters, I assigned colours to those clusters by finding the mean RGB for all pixels falling into that cluster. I have also taken the mean RGB of all outlier points and assigned them that colour. It leads to some noise, yes, but I did it for the sake of simplicity.
For improving the quality of segmented image, one can most definitely use an approach like having a bandwidth and assigning all outliers the cluster group closest in distance to it within the bandwidth. I tried to include variations in the results I obtained. The epsilon, minpts pairs I used were these: `[[0.025, 10]; [0.05, 10]; [0.08, 20];[0.1, 30]; [0.02, 3]]`

The results I got corresponding to these values were:





We can see the effect changing epsilon and minpts has over the clusters. When epsilon=0.1, and minpts required to make a cluster centre=30, due to the large search radius, most points ended up being classified into the same cluster. For points that didn't fall into the main cluster, they tried to form a new cluster centre which they probably did, but it can be possible that due to a high minpts value, many of the outliers remained unclassified and just took a colour equal to their mean value.
For epsilon 0.05 and minpts=10, the plane is the most conspicuous and more segmented, slightly lesser for epsilon 0.08 and minpts=20. This could again be because of the difference in search radius. Note that when the search radius was smaller, i.e. 0.025 and 0.02, we obtained a far more clearly segmented image with a higher number of clusters. The minpts value was also not too high, so only closely resembling pixels became grouped into a cluster and it also helped that minpts was comparatively low which enabled formation of new cluster cores else most of the points at such a low eps value would have remained unclustered and appeared as noise.
The brownish noise in image 1 and 5 is due to the unclassified clusters taking their aggregate mean colour. This can of course be improved and the noise can be

reduced by the method already mentioned at the start. The subtle interplay of the search radius and the min points required to form a cluster core produces different outputs.

3.

Improvements made over SIFT in SURF:

- Instead of gaussian averaging the image as in SIFT, SURF uses convolution of squares on integral images which is much faster and less costly to calculate as they approximate second order gaussian derivatives and can be evaluated very fast using integral images irrespective of size.
- Filter responses are normalised w.r.t the mask size. This ensures that the Forbeius norm is constant for any filter size.
- For finding scale spaces, SURF does not have to apply the same filter again and again to the output of a previous filtered layer since it applies such filters of any size directly on the original image. Thus, instead of convoluting the same filter on downsampled images, SIFT, SURF simply up-scales the filter.
- Since the ratios of filter layout remain constant after scaling, the approximate gaussian derivatives also scale. Forbeius norm being constant ensures that they are already scale normalised.
- Key points are detected using Hessian matrix and Non Maxima suppression.
- SIFT's distribution of gradient features and approximate localised information yields good results as descriptors. Using relative strengths and orientation of gradients reduced effects of photometric changes. But in SURF, the complexity is even reduced. After the interest points (key points) are detected using fast hessian detector,
    - A sliding Orientation window finds the dominant orientation of the Gaussian weighted Haar Wavelet responses at every sample point within a circular neighbourhood around the keypoints.
    - An orientation quadratic grid with 4x4 square sub regions is laid over the interest points for finding descriptors
- Sign of laplacian further differentiates black blobs from light ones, uses no computational cost and in the matching state, can be employed to compare features having the same contrast only. This also gives an increase in performance.

On running the code, SURF was found to be crudely three times faster than SIFT. The outputs are shown below:

```
SIFT:
Elapsed time is 0.151892 seconds.
SURF:
Elapsed time is 0.061104 seconds.
>> q3_2019253
SIFT:
Elapsed time is 0.043981 seconds.
SURF:
Elapsed time is 0.015221 seconds.
>> q3_2019253
SIFT:
Elapsed time is 0.040020 seconds.
SURF:
Elapsed time is 0.016563 seconds.
>> q3_2019253
SIFT:
Elapsed time is 0.043610 seconds.
SURF:
Elapsed time is 0.011065 seconds.
>> q3_2019253
SIFT:
Elapsed time is 0.043952 seconds.
SURF:
Elapsed time is 0.010727 seconds.
>> q3_2019253
SIFT:
Elapsed time is 0.044698 seconds.
SURF:
Elapsed time is 0.007942 seconds.
>> q3_2019253
SIFT:
Elapsed time is 0.042505 seconds.
SURF:
Elapsed time is 0.012561 seconds.
```

For the given image, SURF returned 65 keypoints and SIFT returned 207 keypoints.

I have saved two images in the output folder titled: q3_sift_des.png and q3_surf_des.png that has the top 10 strongest descriptors found using SIFT and SURF respectively, overlaid over the grayscale image input.