CV Assignment 3

Name - Utkarsh Dubey
Roll No - 2019213

Question 1 -

After reading the image I converted all the pixel values in the range of 0-1 by dividing by 255. then took some number of segments.

```
numOfSegments = [50,100,150,200,250]
```

and applied SLIC to obtain cluster labels

```
for num in numOfSegments:
    segments = slic(image, n_segments = num, sigma = 5)
```

Further calculated the number of pixels and values of BGR and then applied the given formula for calculating saliencies

```
for i in range(superPixels.shape[0]):
    saliency=0
    for j in range(superPixels.shape[0]):
        saliency+=formulaCalculation(representativeColors,representativePixels,i,j)
    allSaliency.append(saliency)
```

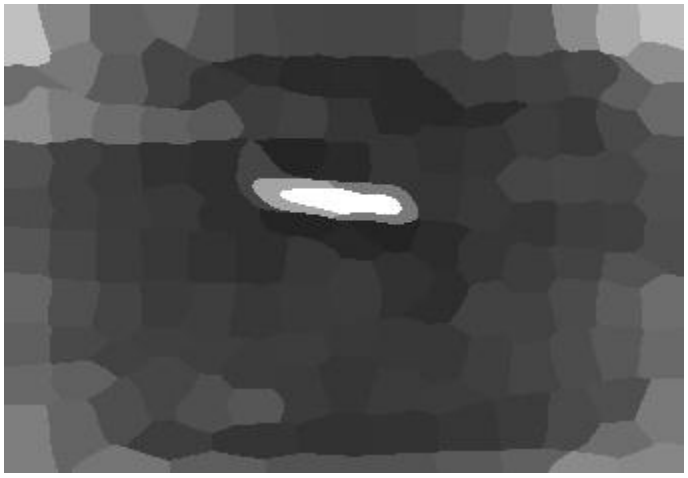
For the number of segments > 100 we can observe decent results.

outputs -

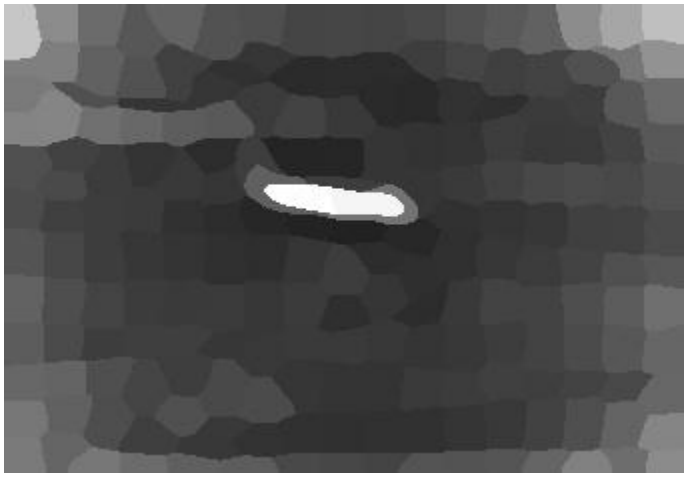
For a number of segments = 50

For a number of segments = 100

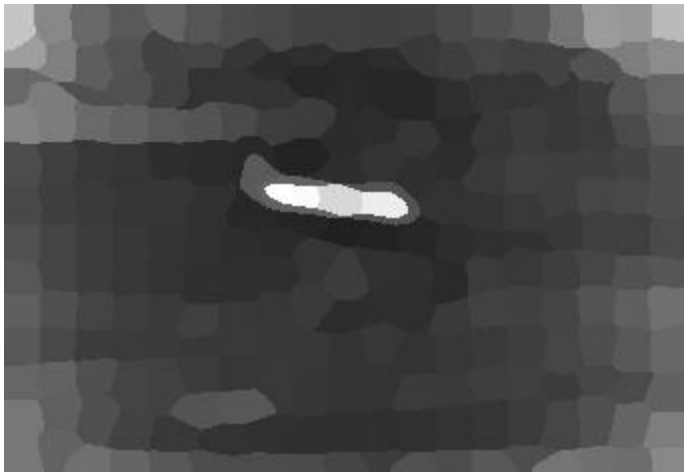
For a number of segments = 150



For a number of segments = 200



For a number of segments = 250



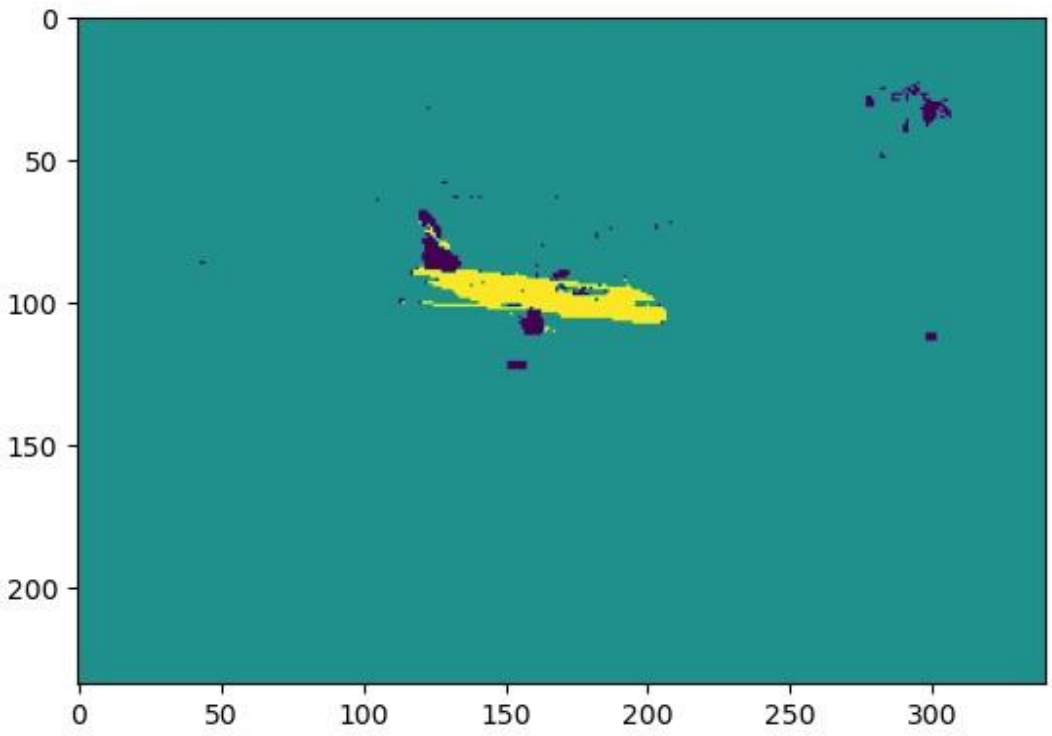
Question 2 -

In code firstly I resized the image to rbgxy space in order to fit the DBSCAN function and then ran it for different parameters

```
for e in [4,8,12,14]:
    for samples in [10,20,30]:
        db = DBSCAN(eps=e, min_samples=samples, metric = 'euclidean', algorithm ='auto')
        db.fit(newImage)
```

As the radius to discover neighbours expands, increasing the eps(radius) minimises the number of outliers. Furthermore, the number of outliers increases as the number of minimum samples required to define a region grows.

outputs -
All outputs in the question2 folder in the outputs folder, images are named for the parameters used.
Best I got for epi - 8 and min_samples = 20



Question 3 -

Code for SIFT

```
def sift(image):
    sift = cv2.xfeatures2d.SIFT_create()
    keypoints, descriptors = sift.detectAndCompute(image,None)
    return
```

Code for SURF

```
def surf(image):
    surf = cv2.xfeatures2d.SURF_create(600)
    keypoints, descriptors = surf.detectAndCompute(image,None)
```

```
time elapsed to run sift -  0.5869464874267578  seconds
time elapsed to run surf -  0.22736287117004395  seconds
```

As we can see SURF took less time than SIFT, Hence SURF is faster than SIFT.

What makes SURF faster is the feature extraction portion of it which is sufficiently efficient when compared to SIFT.

The improvements made in SURF are -

- SURF is based on Hessian Matrix and it takes less time for calculations as it is based on integral pictures and hence is also known as the fast hessian detector.
- In SURF only 64 dimensions are used hence minimising the time for feature calculation and matching.
- We also have a new indexing algorithm in SURF.
- Box filters are also used in SURF for convolution.