REPORT

Aryan Jain *CV Assgn3* *2019101056*

1) **The number of iterations of GMM updating and Energy Minimization:**

   As the number of iterations increases, the segmentation of foreground objects becomes better, but converges after some time (keeping other parameters constant). In my case 5 iterations worked pretty well to get the descent segmentation of the image.

2) **The number of mixture components in your GMM:**
   Increasing the number of Gaussians to fit the image (value of n_components) gives better output as it helps in learning and representing a wider range of colours for the foreground and background. If the number of components is very low, then it may lead to incorrect categorization of many foreground pixels, if the foreground objects bear a wide variation in colour combination. At the same time if there are too many components there can be additional background pixels present in the segmentation of foreground object. Experimentally n_components = 5 works pretty good, but it stills depends on data to data. I took n_components = 5 for my case.

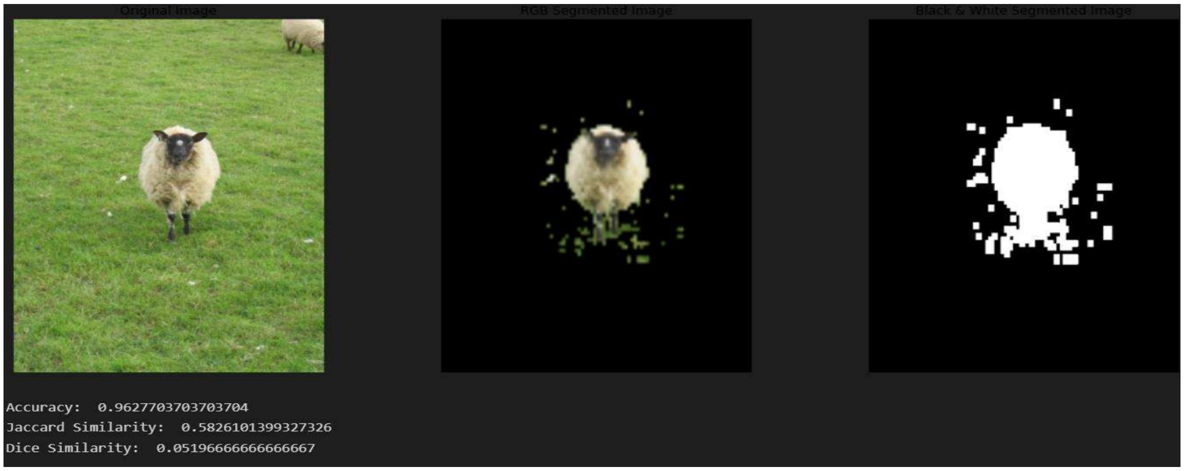
3) **The choice of gamma:**
   Pairwise potential values between any 2 pixels depends on the value of gamma. It is determined experimentally. A too low or too high value of gamma can lead to wrong categorization of the foreground pixels. A value in range of 50-100 works pretty well. For my case I took value of gamma to be 50 as proposed in the research paper.

4) **Different ways to represent probabilities other than GMMs. 4-neighbourhood or 8-neighbourhood in pairwise term:**
   Number of neighbourhoods were changed from 4 to 8 and it resulted in more continuous segmentation as the diagonal pixels would also influence the segmentation. (Just change the eight_connectivity flag to True in the code to run for the 8-connectivity).

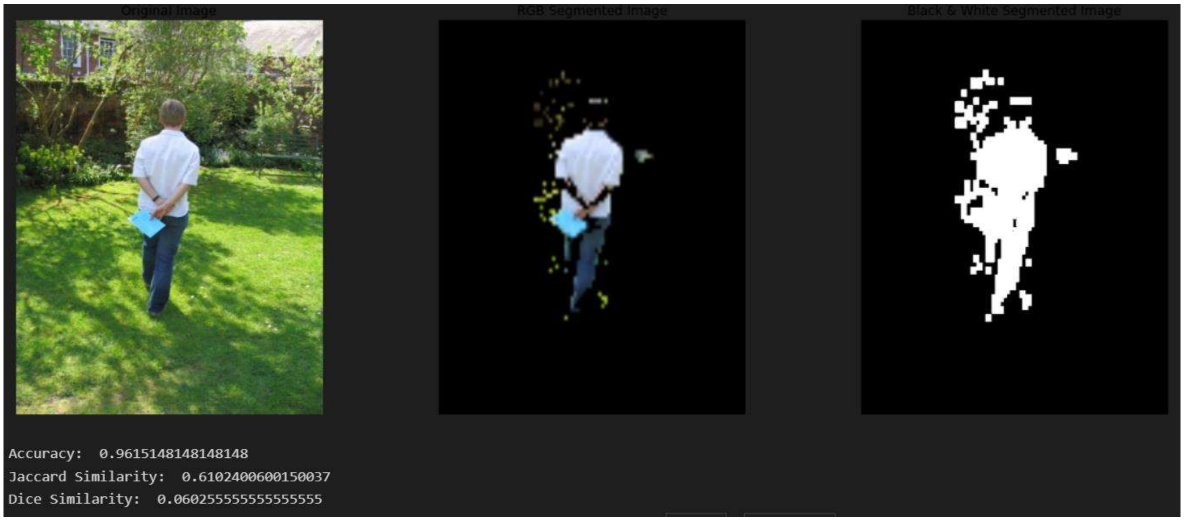
5) **Effect of a tight initial bounding box or loose initial bounding box:**
   If a tight initial bounding box is chosen, then it results in a more stricter supervision i.e. a more accurate segmentation of the foreground object from background. Whereas in the case of loose bounding box a lot of background pixels would be present in region of interest because of weak supervision and which wouldn't be learnt as a part of background model and hence an inaccurate segmentation.

- Time taken to run segmentation on images varied between 60s to 240s.

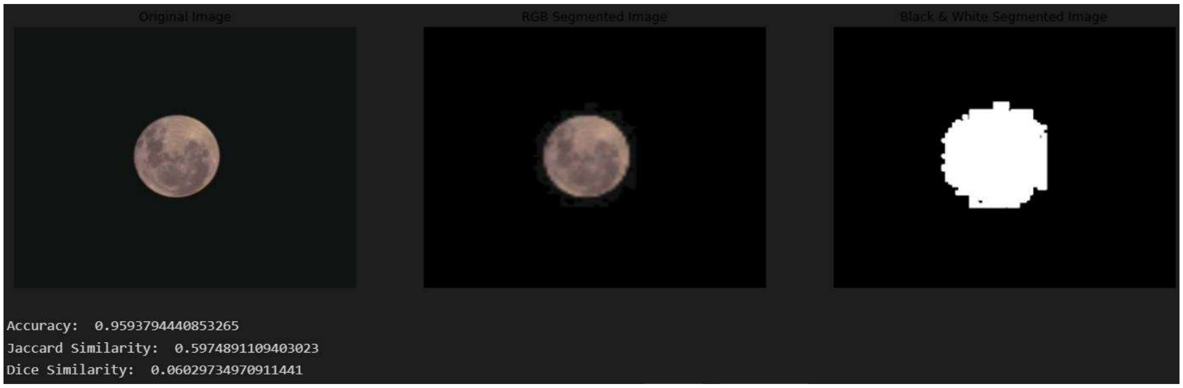
The Algorithm performed well on these images:

1) 

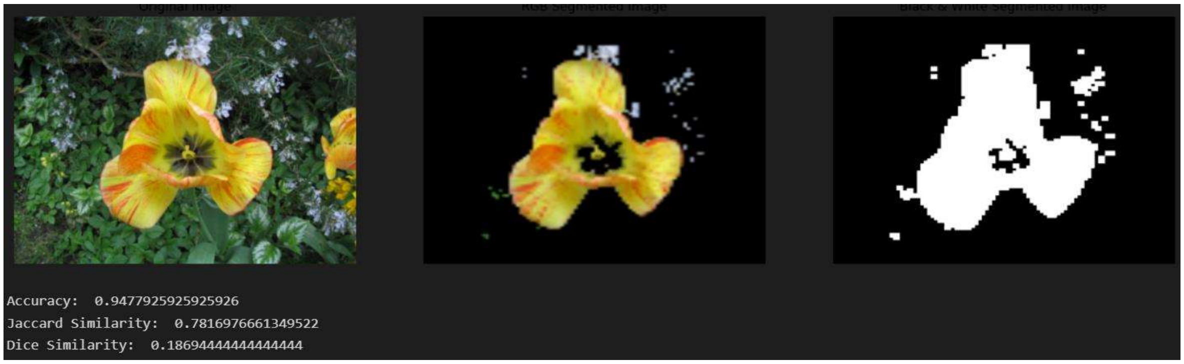
```
Accuracy:  0.9627703703703704
Jaccard Similarity:  0.5826101399327326
Dice Similarity:  0.05196666666666667
```

2) 

```
Accuracy:  0.9615148148148148
Jaccard Similarity:  0.6102400600150037
Dice Similarity:  0.06025555555555555
```

3) 

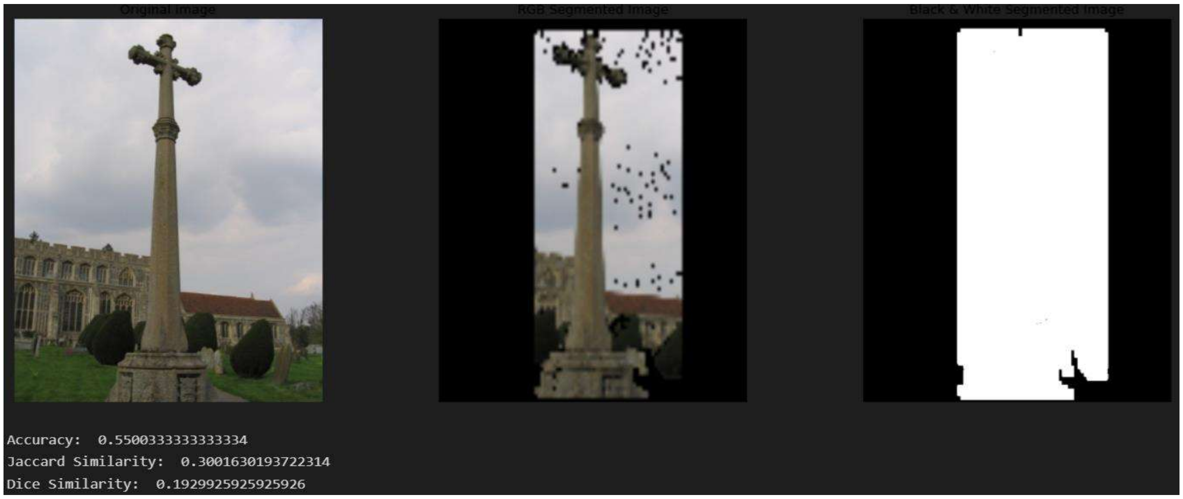
```
Accuracy:  0.9593794440853265
Jaccard Similarity:  0.5974891109403023
Dice Similarity:  0.06029734970911441
```



```
Accuracy:  0.9477925925925926
Jaccard Similarity:  0.7816976661349522
Dice Similarity:  0.18694444444444444
```

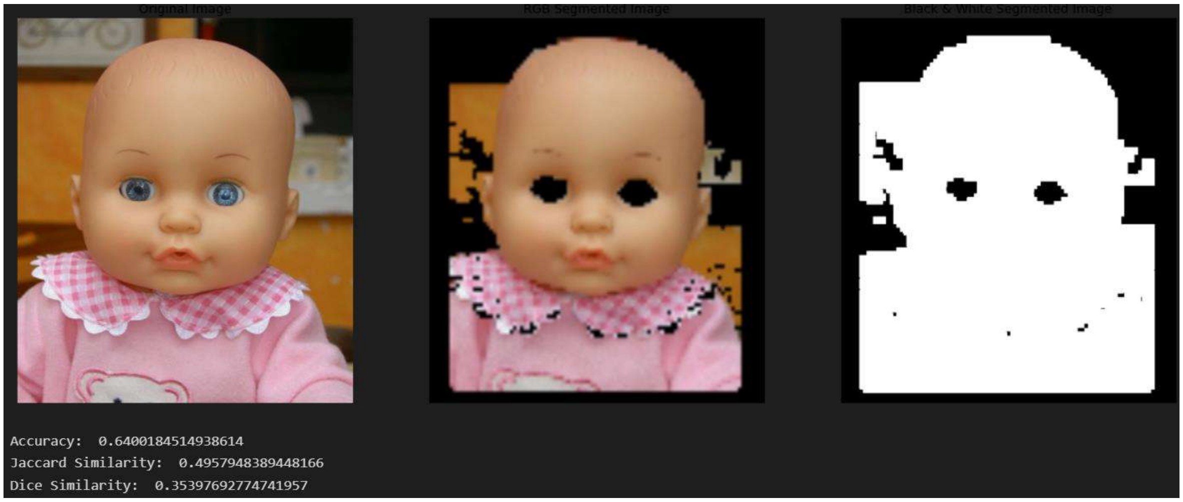
4)

The Algorithm do not have performed on these images:



```
Accuracy:  0.5500333333333334
Jaccard Similarity:  0.3001630193722314
Dice Similarity:  0.1929925925925926
```

1)



```
Accuracy:  0.6400184514938614
Jaccard Similarity:  0.4957948389448166
Dice Similarity:  0.35397692774741957
```

2)