IMPLEMENTATION

- The test images folder inside which images to be segmented are there is uploaded to colab from drive.
- Here two similarity matrices are used to calculate similarity:
    1. Brightness
    2. Colour distance
- Brightness – It is the exponent of negative of difference between the averages of the three channels value of two pixels.

$$e^{-\frac{|avg(pix_1)-avg(pix_2)|}{\sigma}}$$

- Colour distance – It is the exponent of negative of Euclidean distance between rgb channels of two pixels. By Euclidean distance of rgb channel I mean square root of sum of squares of difference between each channel of 2 pixels.

$$d = \sqrt{(r1-r2)^2 + (g1-g2)^2 + (b1-b2)^2}$$

$$e^{-\frac{d}{\sigma}}$$

Where r,g,b are pixel channel values

- The images which has rows or column size more than 100 is down sampled or resized to bring it within 100 or else it takes a lot amount of time to observe the output. For instance image 2 with 132 x 130 size took around 50 minutes to give output without resizing while it took only 7 minutes after resizing by a factor of 2.
- Two functions are created. One that takes 2 pixels and returns the brightness similarity between them and the other takes the pixel values and returns the colour distance similarity between them.
- For each images given the following steps are repeated:-
    1. The weight matrix of ‘pixel x pixel’ size is formed with input from above two functions described.
    2. Then degree matrix is formed (a diagonal matrix).
    3. Then eigen values and eigen vectors are found for the following equation on the basis of which segmentation is done.

$$(D-W)y = \lambda Dy$$

- Following are the resize factors:
    1. Image 1 is not resized
    2. Image 2 is resized by a factor of 2
    3. Image 3 is resized by a factor of 10
    4. Image 4 is resized by a factor of 10
    5. Image 5 is resized by a factor of 10

The following are the real images (after resizing/down-sampling) on which segmentation is done:

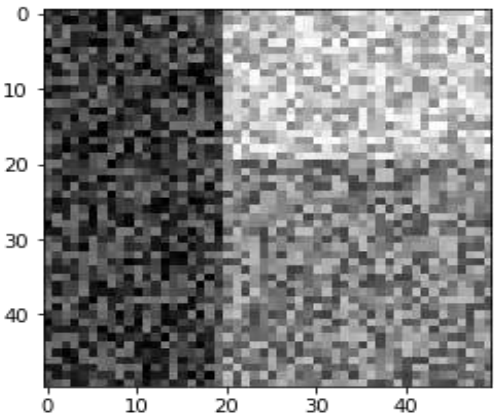

Image 1

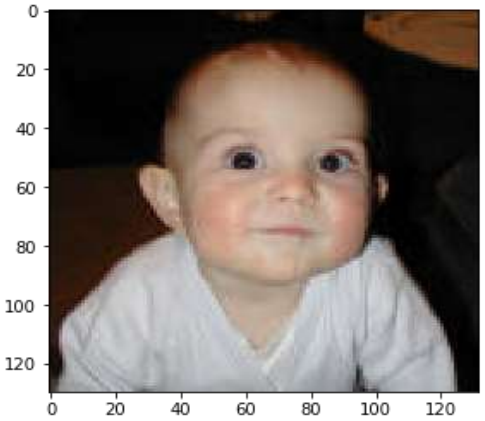

Image 2

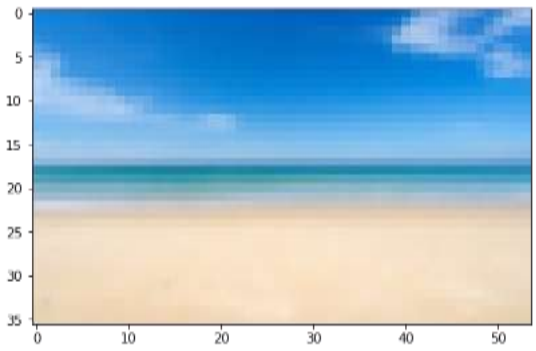

Image 3

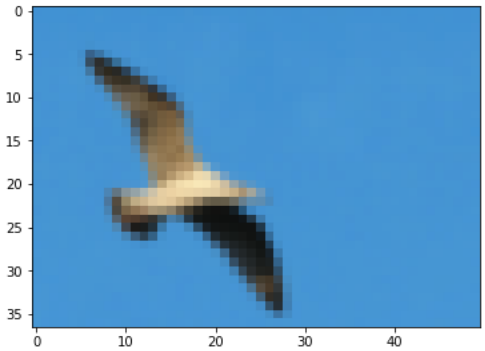

Image 4

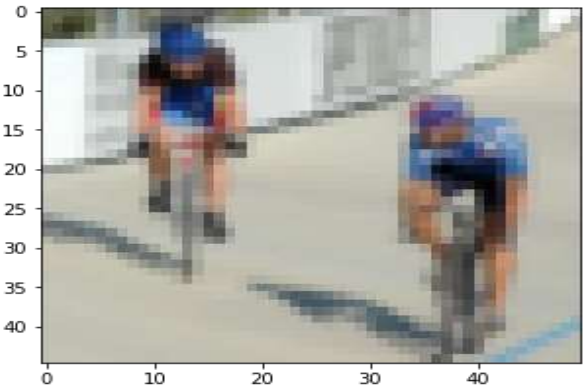

Image 5

With considering only brightness as similarity measure we get the following segmentation as results:

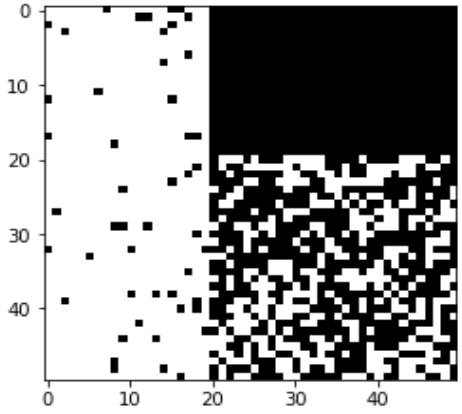

Image 1

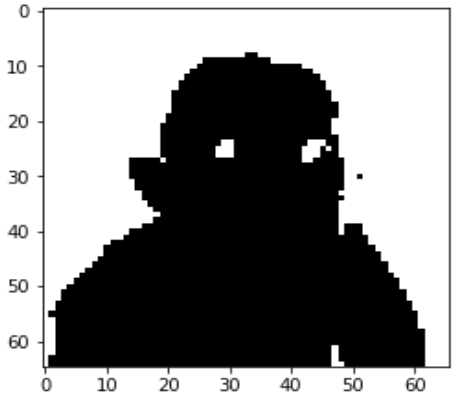

Image 2

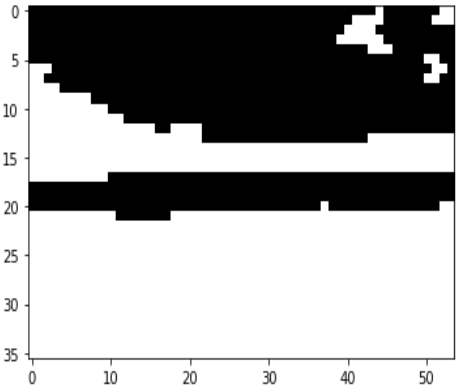

Image 3

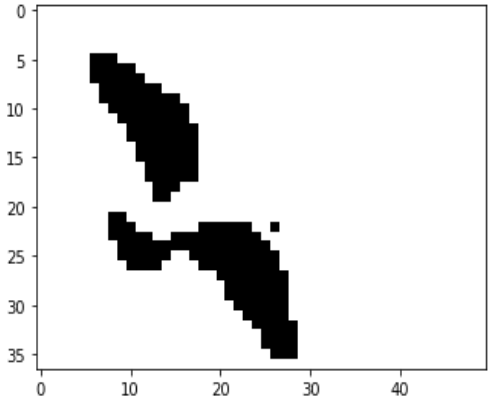

Image 4

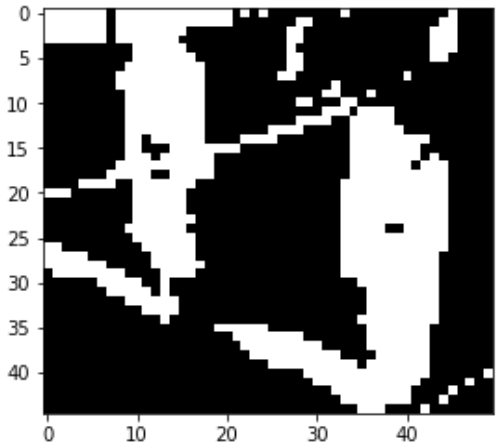

Image 5

Now images are modified and segmentation redone. First modification is rotation. The below images show the effect of rotation on segmentation:

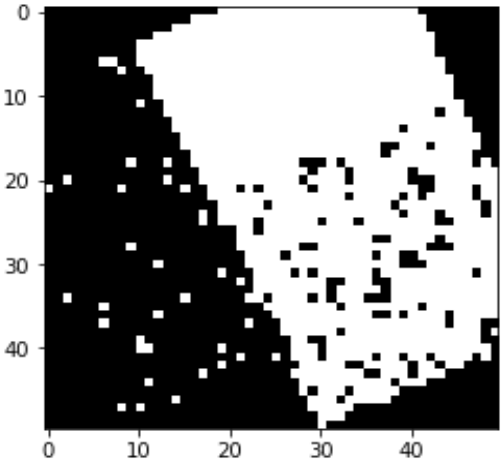

Image 1

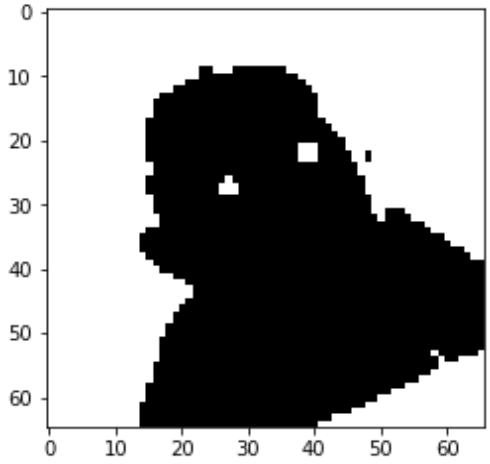

Image 2

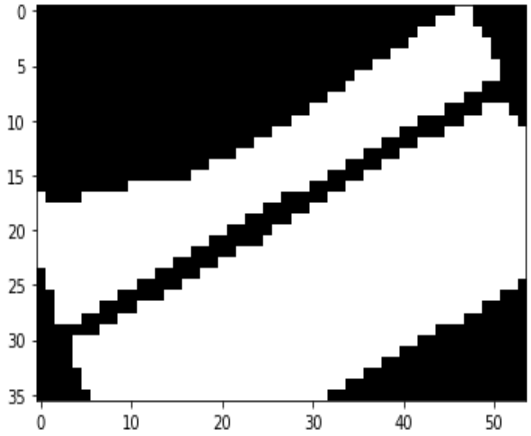

Image 3

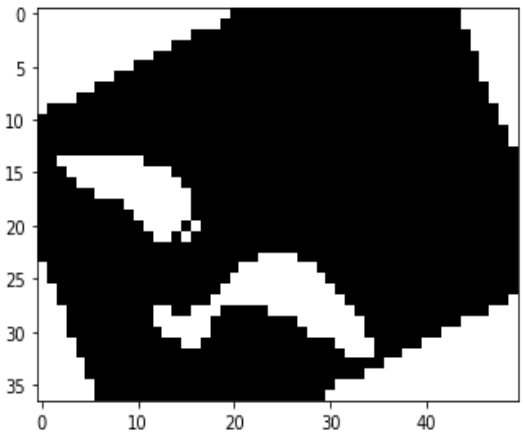

Image 4

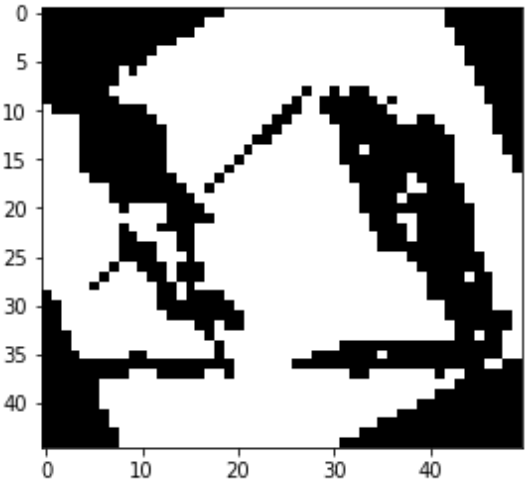

Image 5

Now random Gaussian noise is added to images and then segmentation is done to get the following results:

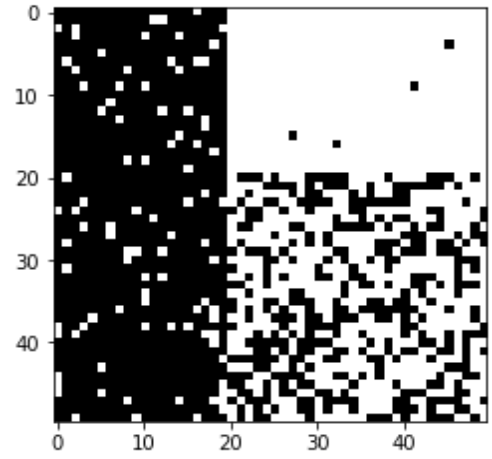

Image 1

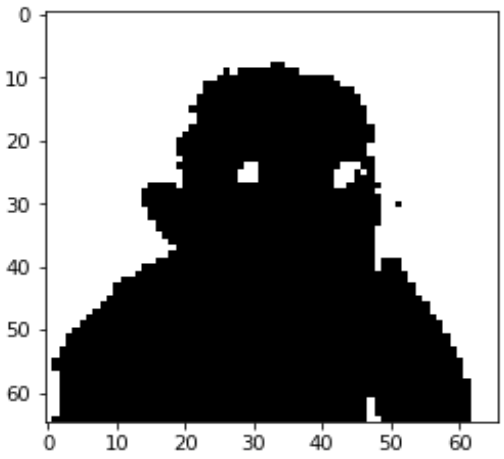

Image 2

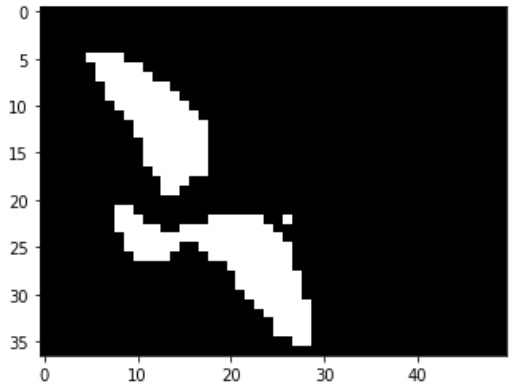

Image 3

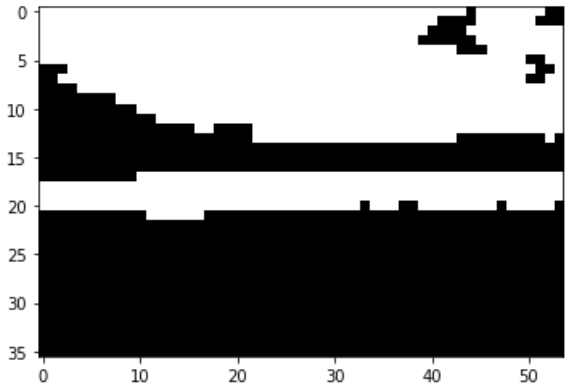

Image 4

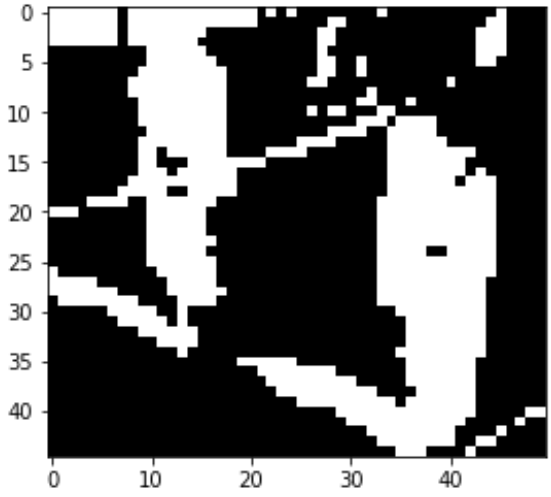

Image 5

Now, with considering only colour distance as similarity measure the segmentation is done:

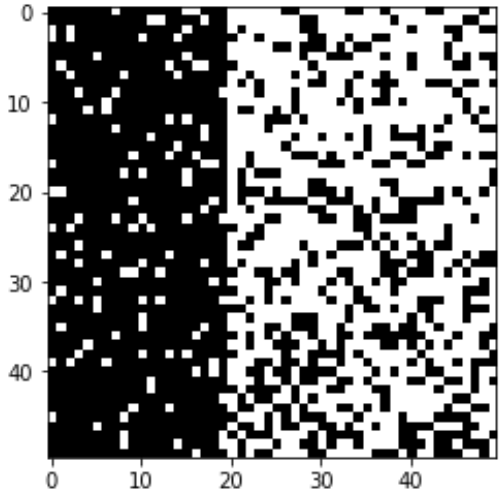

Image 1

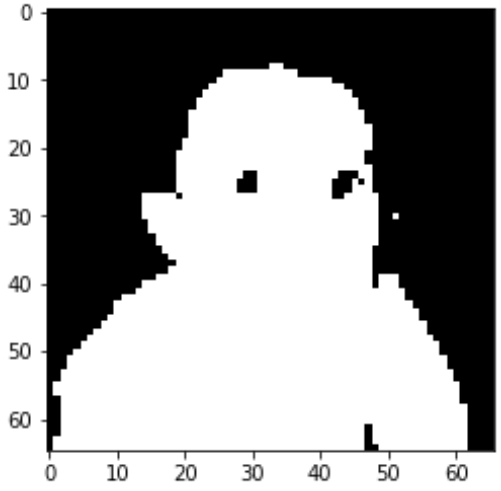

Image 2

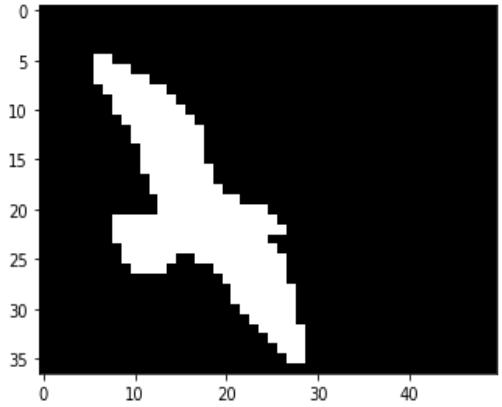

Image 3

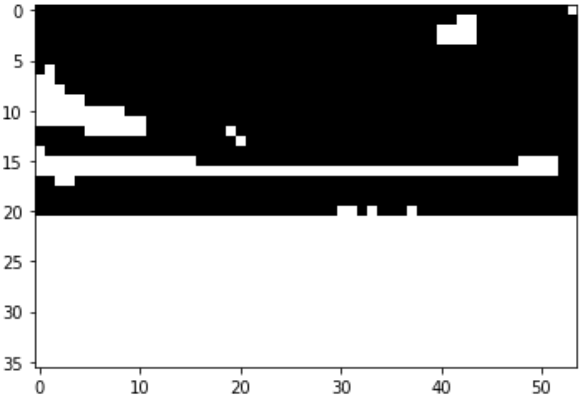

Image 4

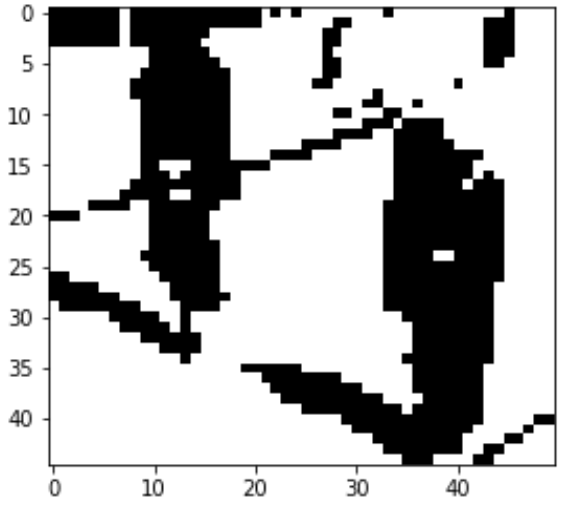

Image 5

RESULT ANALYSIS

- Both brightness and colour distance are good measure for segmentation as both of them were able to segment each image to a good extend.
- There is still few difference between the segments segmented by brightness and colour distance. For instance in the bird image (image 3) colour image was able to segment it properly than brightness. While for image1 brightness is a comparatively good measure.
- As observed rotation and Gaussian noise did not have much effect on segmentation. The only effect of rotation is that the boundary of rotated images also get segmented else everything else remains same.

PROBLEM FACED

- At first I was also considering Euclidean distance as measure along with other similarity measures but it was giving erroneous outputs of only a black and white patch separated by a line. During considering Euclidean distance the parameter ‘r’ has to be set every time an image is changed else erroneous output as mentioned above is faced.
- At first I was not down sampling the image, hence it was taking a lot of time for a single image to get segmented. After that large size image are resized to reduce time taken for segmentation.
- I also tried using cosine similarity measure for segmenting but this did not work out for this kind of images and was giving erroneous results, hence colour distance measure was used instead of cosine similarity measure.
- I had also tried using some other eigen value functions but they were taking a lot of time, so I finally settled on ‘eigsh’ function of ‘scipy’ library.

Code - https://colab.research.google.com/drive/1L4zfoGd76qUPfB3EWt8mvX-wc5sb4RlT?usp=sharing