REPORT

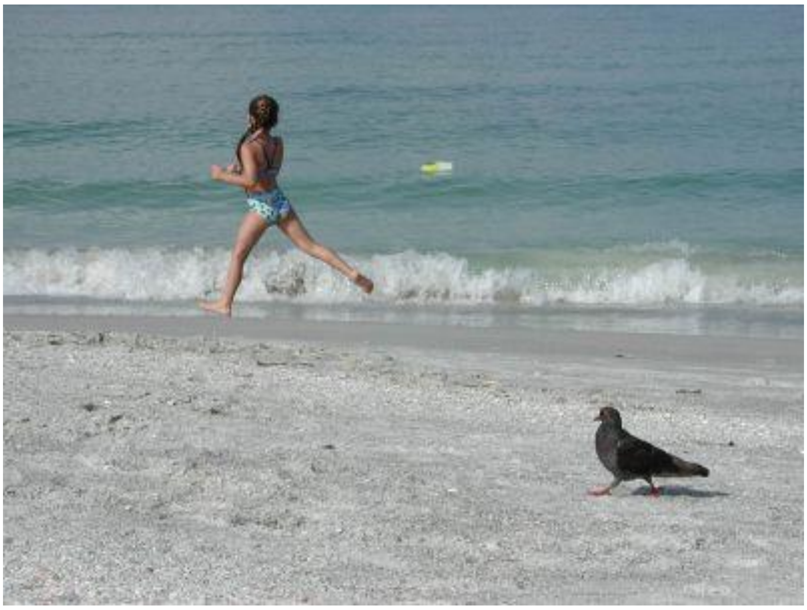
COL783-ASSIGNMENT 3

Ravindra Abhay Kudchadkar 2018TT10941

Abhinav Gupta 2018TT10868

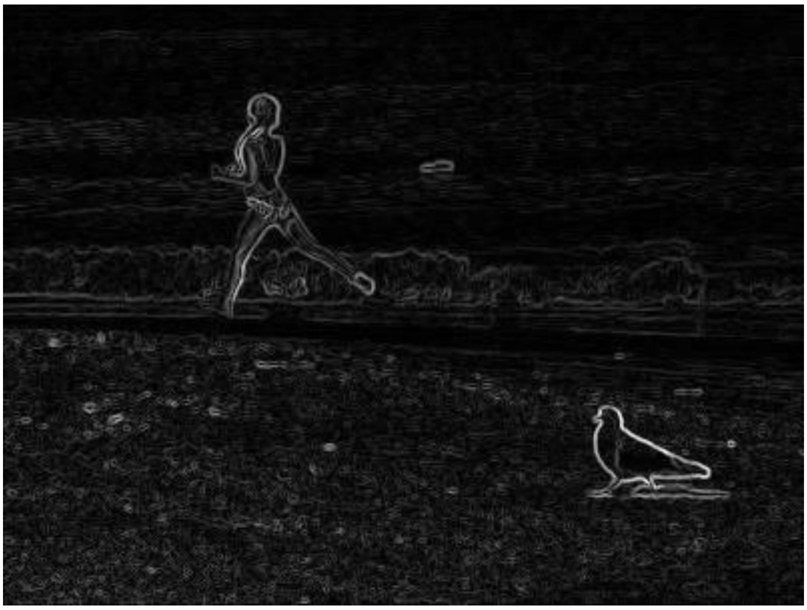
----SEAM CARVING: - CONTENT AWARE IMAGE RESIZING----

Original image:



Energy functions:

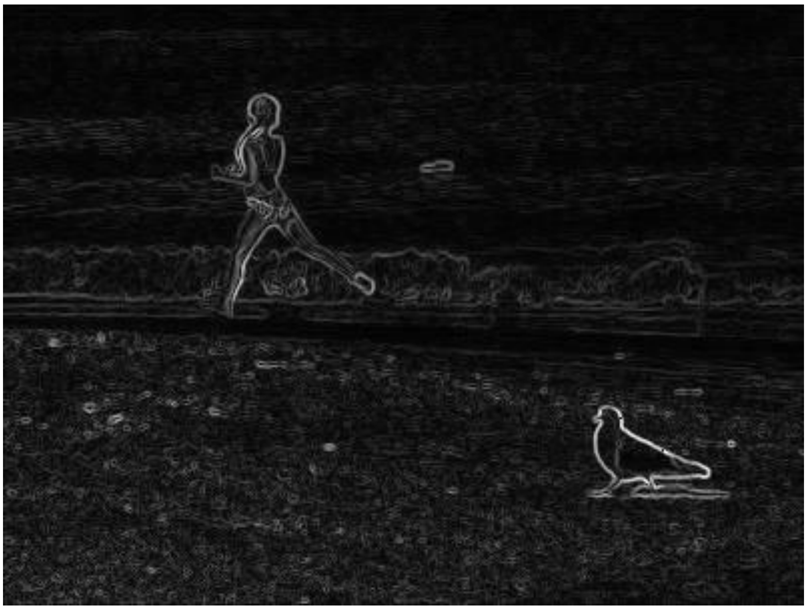
1: E1 gradient: Magnitude of the Sobel operator derivatives in the x and y direction on the grayscale image.



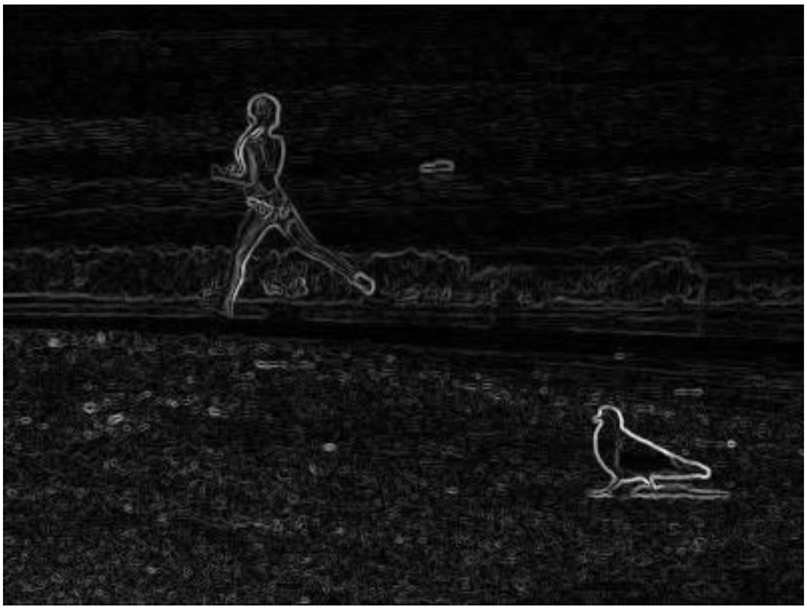
2: Entropy: Computes the entropy over a 9×9 window for the whole image and adds it to E1 energy. Entropy is defined as

$$H(s_m) = -\sum_{n=1}^{256} p_n(s_m) \cdot log_2(p_n(s_m)), m = 1, ..., M$$

Where $p_n(s_m)$ is the frequency of the pixel intensity level in that window of the image.



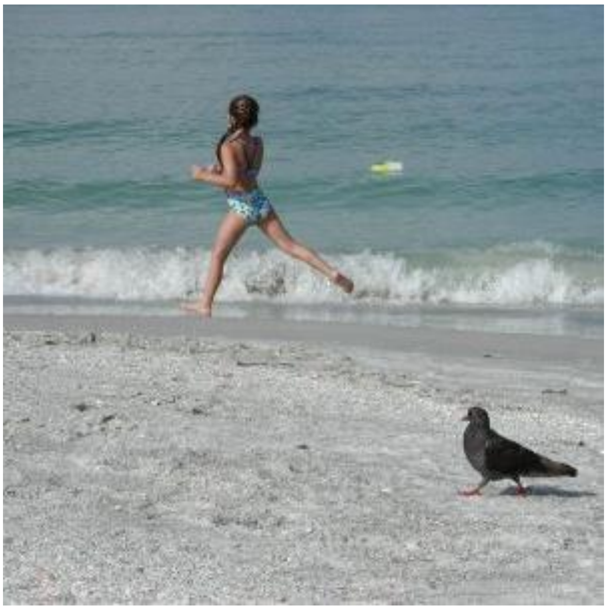
3: E1-HOG: This is E1 energy divided by the maximum of the histogram of oriented gradients computed over a 11x11 window using an 8-bin histogram.



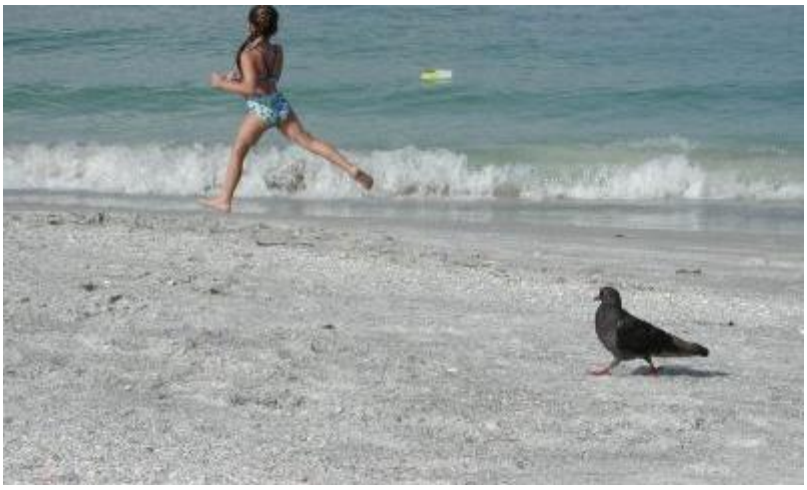
We have used E1-HoG for all the remaining experiments, as we found that this was most efficient.

Seam Removal – Size reduction along a particular axis

Used the dynamic programming formulation to find the minimum seam along that particular axis and then removed that seam.



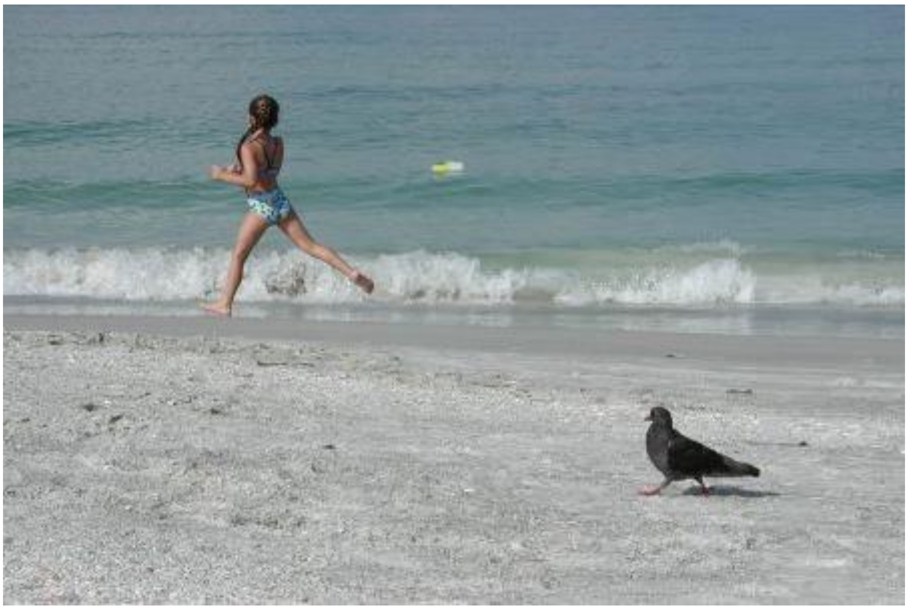
Vertical seam removal to reduce size of image along x axis (100 seams removed)



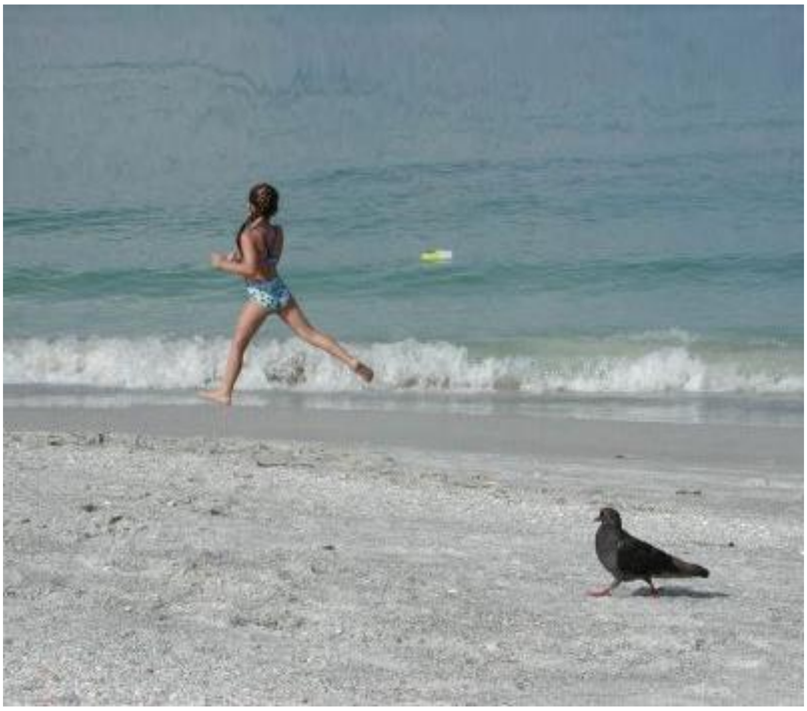
Horizontal seam removal to reduce size of image along y axis (60 seams removed)

Seam Insertion: Image enlargement along a particular axis

Find the minimum energy seam and insert the seam to the right by taking the average values of the pixels to the immediate right and left.



Vertical seam insertion for image enlargement along the x axis (50 seams inserted)



Horizontal seam insertion for image enlargement along the y axis (50 seams inserted)

Image Resizing: This is done by comparing the energy of the vertical seam and horizontal seam at every step and then removing the one that is minimum.

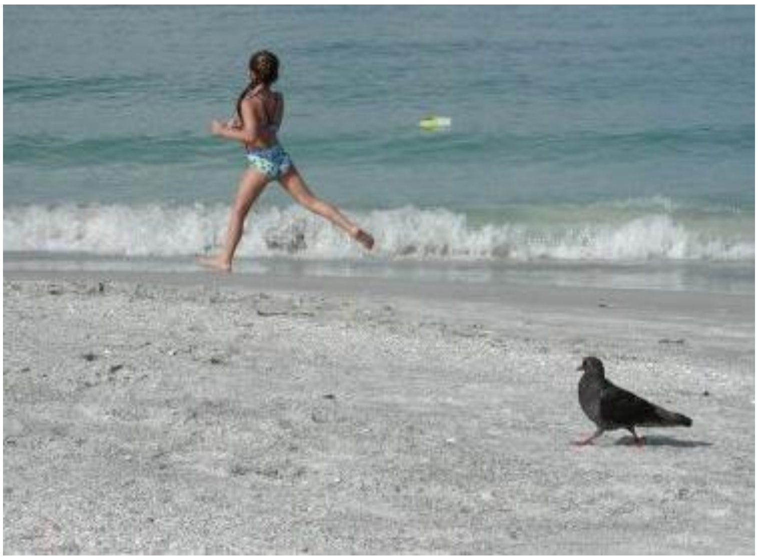
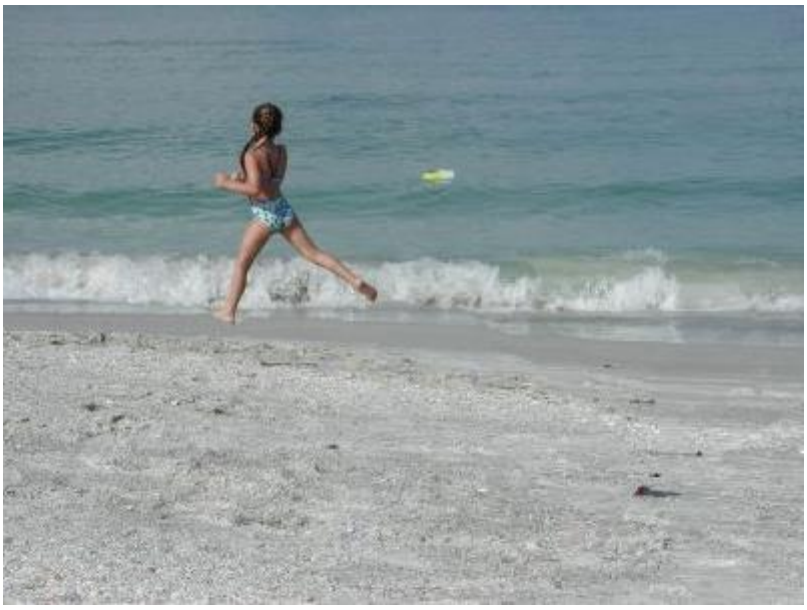
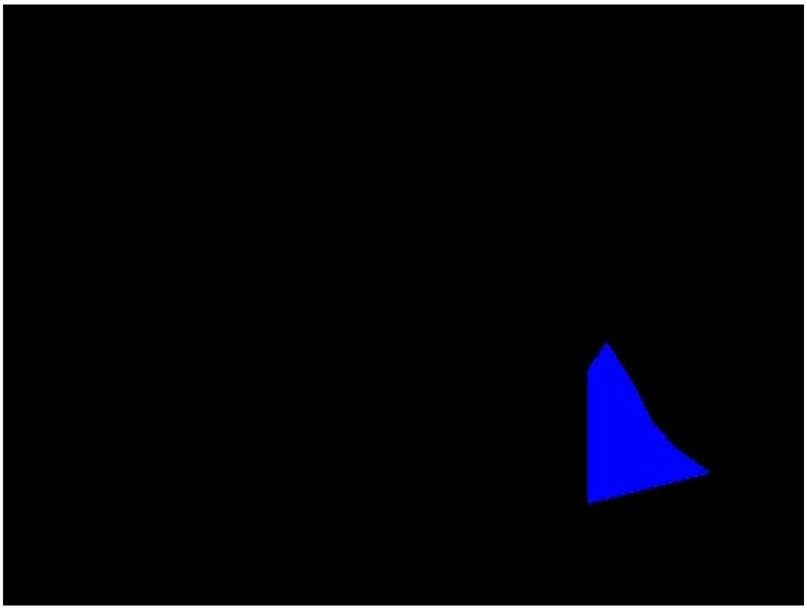


Image resizing

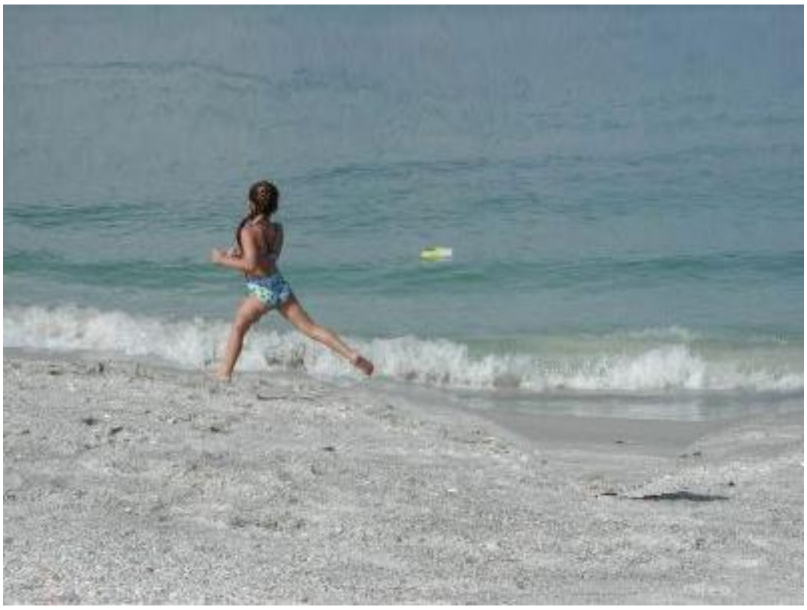
Object Removal: Given a mask for an object in the image, assign the pixels in the mask a large negative value so that the seams pass through it and then keep removing the seams until all the pixels in the mask have been removed. If the vertical length of the mask is more than vertical seams are removed else the horizontal seams are removed.
After removing the object then seams are inserted to get back the size of the original image.



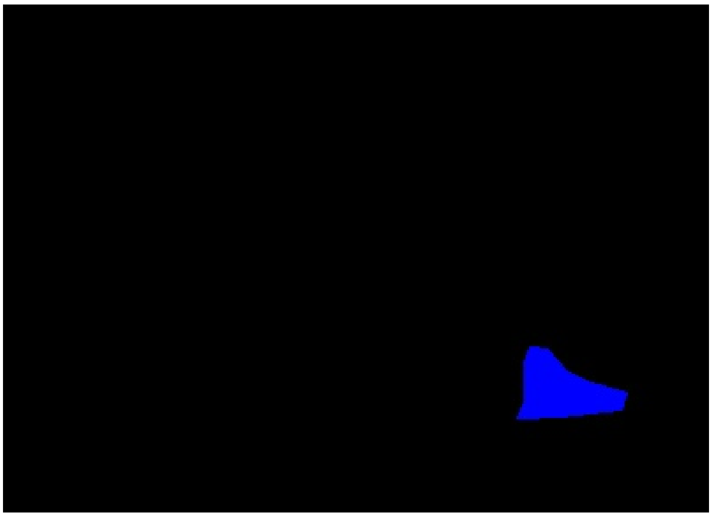
Object removal by removing vertical seams



Mask used

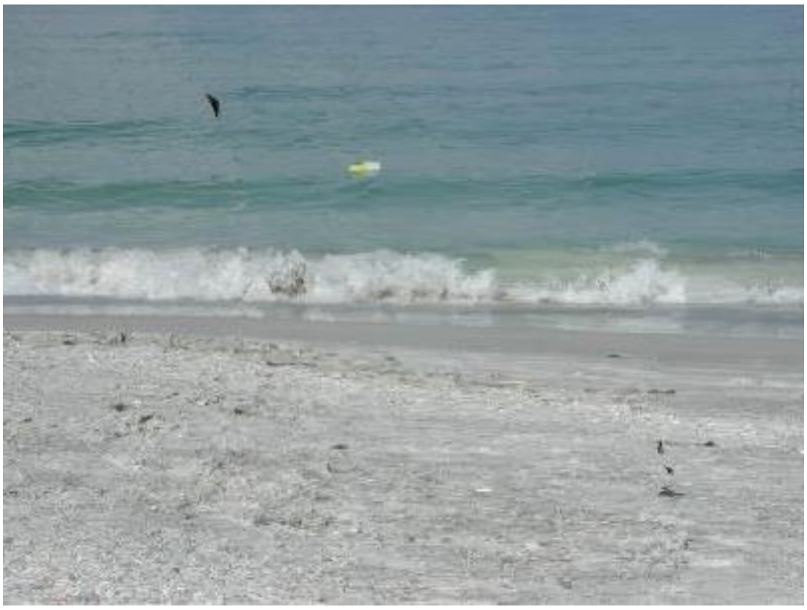


Object removal by removing horizontal seams

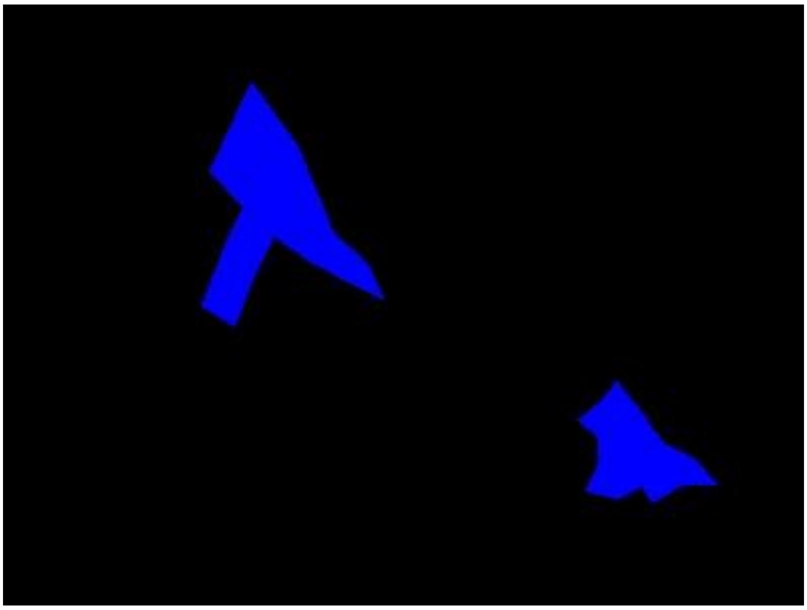


Mask used

Multi-object removal:



2 objects removed and image size is same as the original



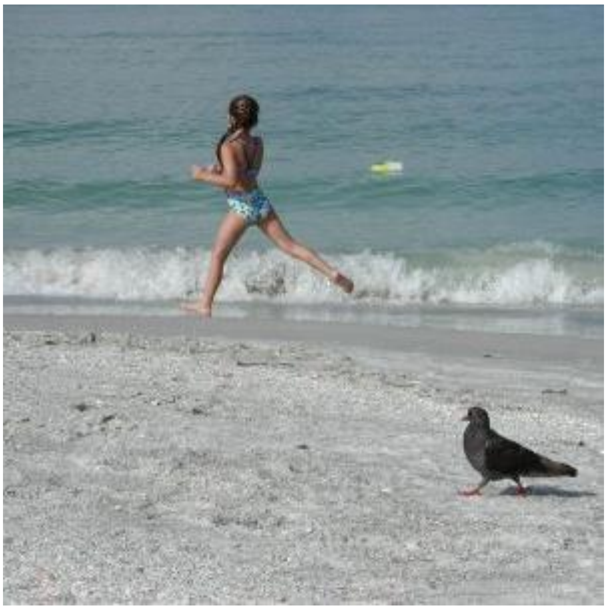
Mask Used

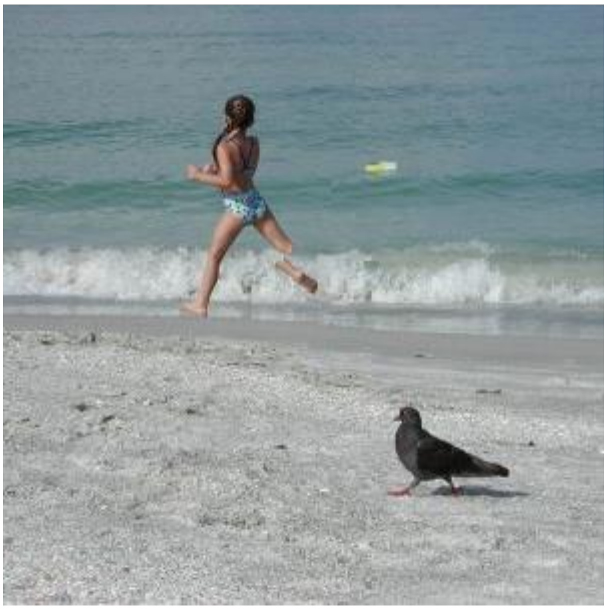
Fast Seam Removal:

Leveraged Gaussian pyramids to increase the speed of seam removal. The algo used was the same as mentioned in the paper.

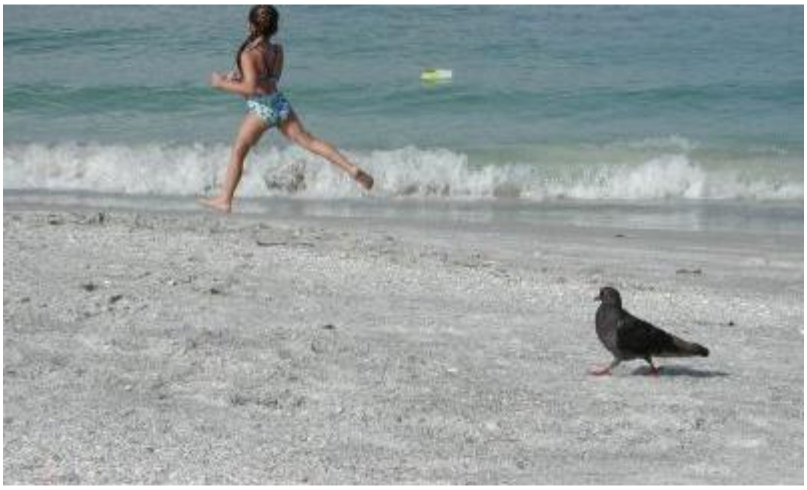
The time taken has drastically decreased but the accuracy of seam removal is compromised. I have used a gaussian pyramid with 3 levels. Removal of 1 seam in the 3rd level is equivalent to removal of 4 seams in the original image.

If the image size isn't even at every step of the pyramid, then I remove the last vertical seam and the insert it again after the image size reduction.


Normal seam removal (vertical)


Fast seam removal (vertical)

The normal seam removal took about 1 minute for removing 100 seams with good accuracy, whereas the fast seam removal took 2 seconds for removing 100 seams but as we can see the result isn't as good as the first one.


Normal seam removal (horizontal)

Fast seam removal (horizontal)

This result for fast removal is almost same as that for the regular one.