Content-Aware Image Resizing

BCSF20A504 - Uswa Tahir
BCSF20A511 - Hamna Suri
BCSF20A513 - Hammad Bin Sajjad

01-03-2023

Introduction:

Content Aware Image Resizing is a technique used to resize an image without the loss of important details. There are many ways to resize images i.e cropping, scaling etc but all these methods can eventually remove the important details from the images. Content Aware Image Resizing prevents the removal of such important content and provides resizing.

At its core, Content Aware Image Resizing using Seam Carving. Seam Carving is a technique which is used to identify the parts of an image that have the least amount of details. Its implementation is provided in the upcoming sections of this report. Seam Carving plays the most important role in Content Aware Image Resizing as it can locate the least detailed parts of an image and we can then remove it reducing the size of the image, preserving the important details.

Now Content Aware Image Resizing is not only effective on image size reduction but also it can be used to enlarge the image. Using seam carving, we find the least details and then adding then increases the size of the image, but preserves the size of the important details and objects. Another interesting use case for this technique is object removal. We could not implement the object removal because of its implementation complexity.

In the following sections, we will discuss the implementation and working of the entire project using multiple images and different use cases.

Literature review:

Research Papers:

We used 2 research papers for this project.

- Seam Carving for Content-Aware Image Resizing [1]
- Content-Aware Image Resizing [2]

Articles and Blogs:

Following articles and blogs were referred to while making this project

- Context-aware image resizing with seam carving [3]
- Content-aware image resizing in JavaScript[4]
- Seam Carving Algorithm : A Seemingly Impossible Way of Resizing An Image [5]

Videos:

Following video resources were used in the understanding and the implementation of the project

- Content Aware Image Resizing (Seam Carving) [6]
- Content-Aware Resizing using Seam Carving [7]

Other Resources:

Other resources used in the project are as follows

- JS Image Carver [8]

Implementation Details:

Language and Development Environment:

Our programming language of choice was Python. Python was suitable for our project because it is widely used for computational programming around the world. Also, libraries like OpenCV, SKImage, Numpy made it extremely easy to work with images without getting too much into the details that were not necessary for this project..

Our development environment was “Google Colab”. Google Colab is an online hosted instance of the Jupyter Notebook. We used it because it offers a collaboration feature in which multiple people can write code in the same jupyter notebook. Furthermore, since it is cloud hosted, Google offers the free of cost use of its processing CPUs, GPUs and TPUs. Since our project required heavy computation for image processing, using Google Colab provided us with a better platform to code, run and test our project in a much easier manner.

Project Division:

Our project was divided into 4 major parts that were then assigned between the 3 team members for easy collaboration and integration of code.

Part 1: Seam Carving (Assigned to Hammad):

The crux of our project lies in the detection of the seam. Seam is a path in the image where there are the minimum details. In order to locate the seam path in the image, we had found 2 approaches.

- Greedy Approach
    - Find the smallest value in the 1st row, then the smallest value in the neighbouring 3 pixels below, and then repeat till the last row.
- Dynamic Programming
    - Traverse all possible paths for the seam recursively and find the path that gives us the minimum energy in total. Many times the program would traverse repeated paths, we used memoization to reduce the running time of our code.

The greedy approach ran faster but didn’t guarantee finding the optimal path to remove from the image, whereas the other DP approach tended to be slower and costly but provided us with the minimal possible seam.

The process of seam carving goes as follows

- For each pixel in the first row, find the minimum path and compare it with the rest, if it’s total value (cost) in smaller than all others, that pixel will be the starting point of the seam
- For the seam starting point pixel, traverse a path visiting all neighbouring nodes below till the last row then backtracking keeping record of the minimum value path. Once completely backtracked we have our minimum value seam

Part 2: Seam removal/addition (Assigned to Uswa and Hamna):

Once the seam is found, the next thing to do is to either remove it from the image to make it smaller or enlarge it by adding the seams.

Seam Removal (Assigned to Uswa):

The remove_seam function takes an image and a seam path as input and returns a resized image with the seam removed. The function first checks if the input image is a grayscale image or a colour image. It then creates a new image with one less column than the original image and the same number of rows and colour channels as the original image.

Next, the function loops through each row of the original image and each pixel in the row. If the current pixel is not part of the seam path, the function copies the pixel value to the corresponding location in the new image. If the current pixel is part of the seam path, the function skips the pixel and does not copy it to the new image.

Seam Addition (Assigned to Hamna):

Seam addition firstly uses seam removal to determine the seams that are to be added in the image. We cannot just directly perform seam addition because that way the seam that we located at first would be selected over and over. Instead what we do is remove seams from the copy of the image and save them in a list. Once all the seams have been removed, we then in reverse order add the seams in the original image. Adding is just essentially copying the values into the right pixel.

The function add_seams takes in the image and the seam that is obtained from the reverse of the removed seams list. And adds each seam into the resized image i.e with 1 additional column. Then this newly resized image is returned for more seams to be added in it (if required).

Part 3: Optimization (Assigned to Hamna and Uswa):

The algorithm for Content Aware Image Resizing is now complete. But it can be further optimised to perform better in under certain cases. One such case that we encountered was the distortion of facial structures and details when removing rows and/or columns.

The paper [1] had provided a solution for this problem. Using some face detection techniques (we used the built in CV2 Library detection function) to detect the rectangle coordinates where the faces of people (frontal faces) are located and in the energy map, set those pixels to zero. Now when seam carving happens, it ignores those pixels because of being very high in values as compared to the rest of the images.

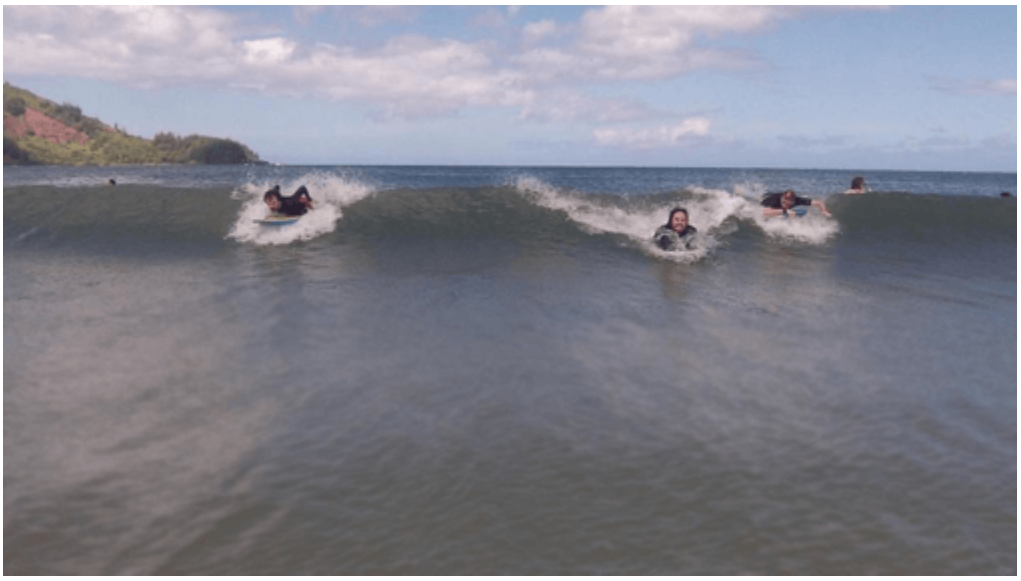
An example of this can be seen in the section below. This drastically improves the working of the algorithm and provides a lot better results as compared to without preserving the faces.

Part 4: Testing and Usage (Assigned to Hammad):

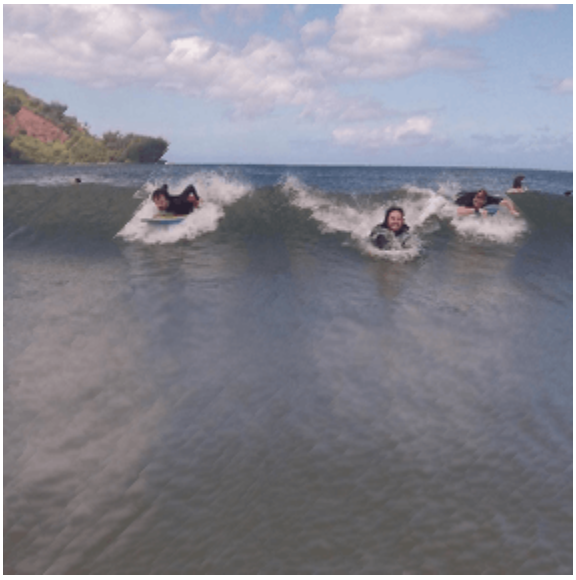
When the implementation is complete, we used different images and performed resizing of different images whether that be column or row removal or enlargement. For more details about the implementation and examples, please refer to the section below.

Results:

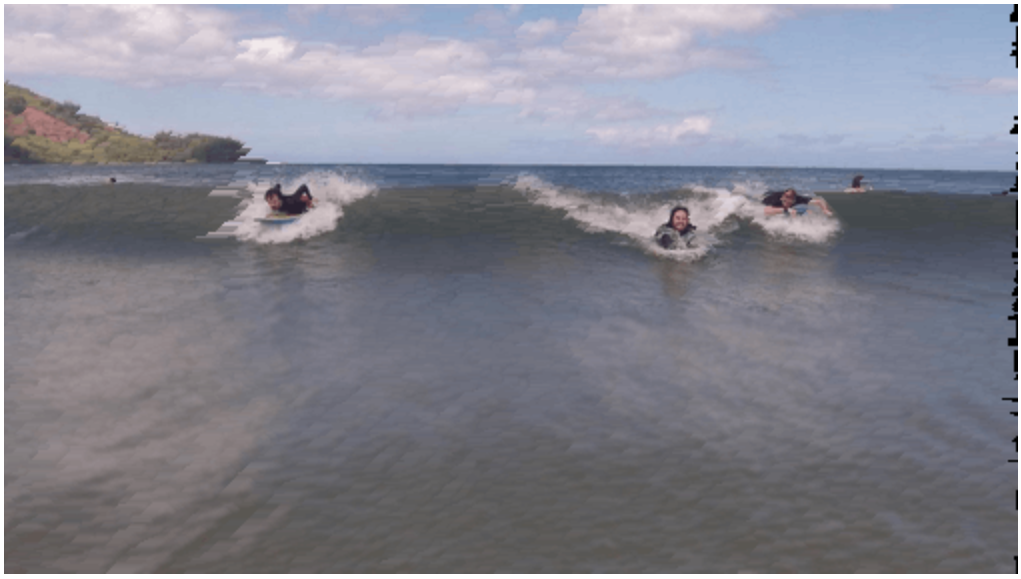
Following are the results obtained before and after the normal resizing operations



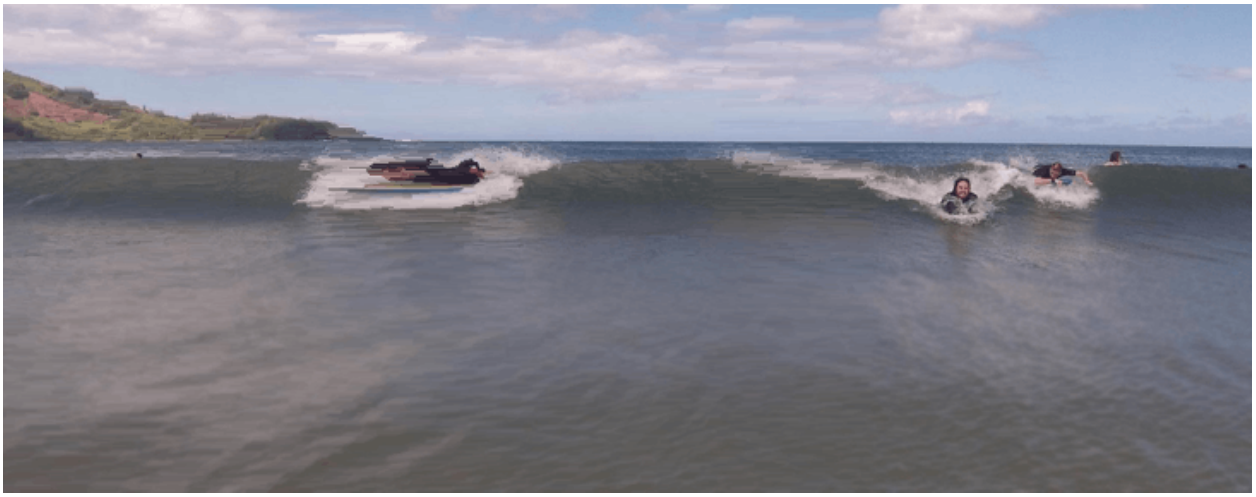
1. Original Image, further used to perform resizing (reduction and enlargement)



2. Reduced Image, removed 222 seams to convert to a square image



3. Reduced Image resized back to the original, slightly distorted at the end because of information loss.

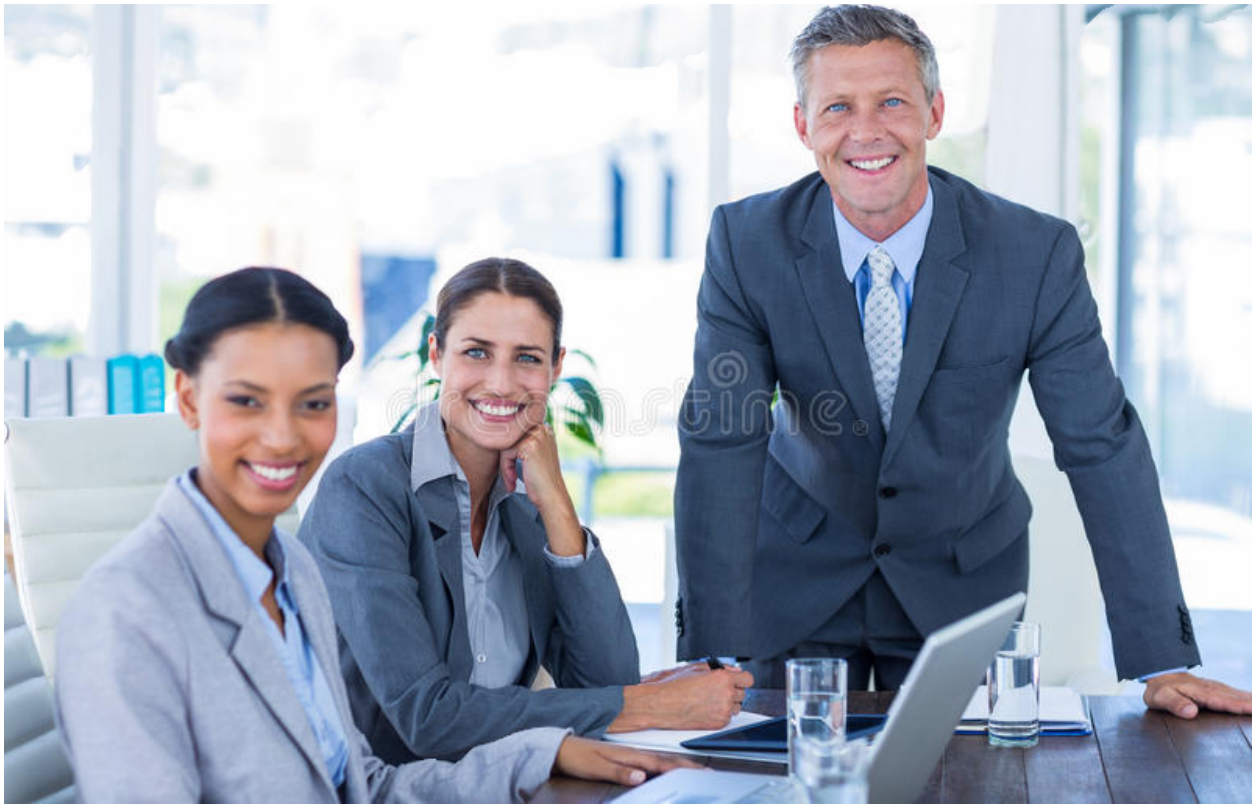


4. Original image enlarged with the seams removed from the image. Stretching can be seen and some points.

Seam Carving can produce undesirable results like face distortion as talked about above. In order to prevent that, we use face detection to highlight them in the energy map. The results of before and after face detection can be seen as follows

1. Original Image, focus on the face of the person standing.

2. The face of the person standing gets distorted when the horizontal seams are removed from the image.



3. Removing horizontal seams, but preserving faces. The face is not distorted and kept its original form.

Conclusion:

Content Aware Image Resizing is a very effective way to reduce or enlarge the image without messing up the details in the image. Now it's not a perfect method to perform resizing as under many different circumstances this technique fails to perform well. In those scenarios it causes more harm to the image than good. Thus this technique should be used when there's a small portion of image that's distinct from the rest. That will provide us with a good amount of seams that pass through the non distinct parts and give us the path that has the least details. Removing such seams doesn't alter most of the important details and thus the quality of the image stays preserved even when it is stretched or reduced.

References:

[1] S. Avidan and A. Shamir, “Seam Carving for Content-Aware Image Resizing.”

[2] P. Zargham and S. Nassirpour, “Content-Aware Image Resizing.”

[3] G. Boduljak, “Context-aware image resizing with seam carving,” Context-aware image resizing with seam carving, Jun. 15, 2020. https://gabrijel-boduljak.com/image-seam-carving/ (accessed Mar. 02, 2023).

[4] O. Trekhleb, “Content-aware image resizing in JavaScript,” Content-aware image resizing in JavaScript, Apr. 16, 2021. https://dev.to/trekhleb/content-aware-image-resizing-in-javascript-1k04 (accessed Mar. 02, 2023).

[5] S. Dash, “Seam Carving Algorithm : A Seemingly Impossible Way of Resizing An Image,” Seam Carving Algorithm : A Seemingly Impossible Way of Resizing An Image, Sep. 15, 2020. (accessed Mar. 02, 2023).

[6] N. Dileas, “Content Aware Image Resizing (Seam Carving),” www.youtube.com, Mar. 23, 2018. https://youtu.be/-YfbEd_48uY (accessed Mar. 01, 2023).

[7] “Content-Aware Resizing using Seam Carving,” www.youtube.com, Mar. 27, 2016. https://www.youtube.com/watch?v=bdCAlUxFyG0 (accessed Mar. 01, 2023).

[8] O. Trekhleb, “JS IMAGE CARVER,” trekhleb.dev. https://trekhleb.dev/js-image-carver/ (accessed Mar. 01, 2023).