Project Report

Abstract

The aim of this project is to regenerate the results of original paper Seam Carving for Content-Aware Image Resizing. Seam-carving is a content-aware picture downsizing technique that reduces the image's height (or width) by one pixel at a time. A vertical seam in an image is a line of pixels connecting one pixel in each row from top to bottom; a horizontal seam is a path of pixels connecting one pixel in each column from left to right. I implemented the base algorithm from the paper and added the feature to increase the height and width of the image.

1 Introduction

The goal of this seam carving method is to do picture scaling that is content aware. This enables for picture resizing without losing important material due to cropping or scaling. The goal is to find the image's ideal seams, which are linked pixel lines that go from top to bottom or left to right, and delete or introduce them while maintaining the image's original features.

The approach detects the lowest-energy "seams" that stretches across the image by first identifying "low-energy" portions of the image that are less intriguing. When lowering the width of a picture, seam carving looks for a vertical seam that runs from the top to the bottom, shifting just one pixel left or right from one row to the next. We minimize the width of the picture by one pixel by detecting and deleting the lowest-energy seam. By repeating this technique, we may significantly minimize the width of the image.

2 Related Work

2.1 Shai Avidan and Ariel Shamir. (2007). Seam Carving for Content-Aware Image Resizing. SIGGRAPH.

This paper originally introduced the algorithm for seam carving and there has been many improvements to the algorithm since then. The authors of the original research propose content-aware picture scaling, which entails altering the width or height of an image while taking into consideration its contents.

There are a variety of approaches to minimize the width of a picture, as detailed in the article. Cropping and scaling is an option, each with its own set of drawbacks, but there's also the option of eliminating columns of pixels from the image's center. As you may expect, this results in a noticeable line in the image where the left and right images don't match up. Even after completing all of this, you can only erase so much of the picture this way. This approach however, works slow for medium or large size images. There has been some modification in the algorithm with the use of recent and advanced libraries which helps in the faster processing even for high quality images.

2.2 Michael Rubinstein and Ariel Shamir and Shai Avidan. (2008). Improved Seam Carving for Video Retargeting. ACM Transactions on Graphics (SIGGRAPH).

The original seam carving paper focuses on deleting seams with the least amount of energy while ignoring the energy put into the photos and video by applying the operator. To combat this, the new criterion looks forward in time, deleting seams that provide the least amount of energy to the retargeted output. There are also certain restrictions to seam carving. Seam carving can cause

substantial artifacts on photos with prominent spatial characteristics. To solve this issue, a new seam cutting criterion has been developed that better maintains important spatial and temporal material. This significantly enhances the visual quality of the retargeted photos and videos.

This paper introduces new techniques such as seam carving using Graph cuts, Graph cuts for videos and Forward energy in Graph cuts. To distinguish them from picture edges, graph edges are represented as arcs. From the image, a grid-like graph is created, in which each node represents a pixel and links to its neighbors. S (source) and T (sink) virtual terminal nodes are formed and linked to all pixels in the image's leftmost and rightmost columns, respectively, with infinite weight arcs.

3 Contribution

To resize pictures in a context-aware way, we must first decide which pixels from the original image should be eliminated. To do so, we must first create an energy map of the image, which we can then utilize to delete unnoticeable, low-energy pixels. To create the energy maps for this project, we used a Sobel gradient energy function with a 3x3 kernel. Whereas, in original paper this formula was used:

$$e_1(\mathbf{I}) = |\frac{\partial}{\partial x}\mathbf{I}| + |\frac{\partial}{\partial y}\mathbf{I}|$$

Figure 1: Original Formula from the paper.

3.1 Reducing Size of an Image

We may now shrink the width of a picture using any energy function. We found the vertical seams in the image with the lowest total energy and then deleted them. Using the following formula, we calculate the summed energy of all potential pathways for each point (x, y) in each row of the picture:

$$N(x,y) = e(x,y)+\min(N(x-1,y-1),N(x-1,y),N(x-1,y+1))$$

This formula takes the energy of the current pixel and divides it by three to get the energy of the three pixels below it (x, y). It then adds the position of the pixel with the lowest energy to the current route, traversing the entire image and forming a vertical seam in the process.

3.2 Scaling up of Images

We wish to just enlarge the image by the lowest energy seams while increasing its size to avoid changing the image's high energy components. While enlarging a picture, though, you can't merely add the lowest energy seam and repeat the process as we did when shrinking it. We must make a temporary replica of the original picture, execute an image reduction on it, and gather all of the lowest energy seams that were eliminated while doing so. Then we went back to our original image and added that collection of low-energy seams.

3.3 Improvements

When I implemented the algorithm initially, it took very long time to process medium or large sized images. Upon searching, I got familiar with a numba library which significantly improved the processing time of this algorithm. To use numba all we have to do is just import library as "import numba as nb" and then write "@nb.jit" before every method we need to use numba for.

3.4 Warnings and Shortcomings

While running the program many Deprecation warnings are generated due to the use to numba library. I tried to supress these warnings with several methods but none of them seems to work. Hence, The submitted code shows these warnings while running it. Another area where I struggled was while trying to get a copy of image saved with minimum seam line across it, an error saying

"array is 0 dimensional but 2 indexes were passed" was raised. The same input was working fine with other methods but I think there was some mistake I was making while transposing the coordinates.

4 Results

4.1 Images with their energy maps

Figure 2: Image of a dog with its energy map

Figure 3: Image of a sunflower with its energy map

4.2 Images with reduced height and width

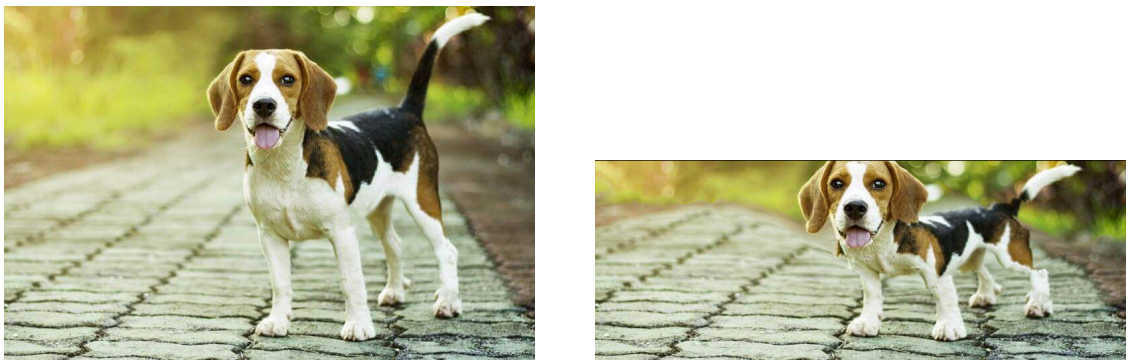

Figure 4: Left Image HxW: 533x800, Right Image HxW: 300x500

Figure 5: Left Image HxW: 400x600, Right Image HxW: 300x300

4.3 Images with increased height and width

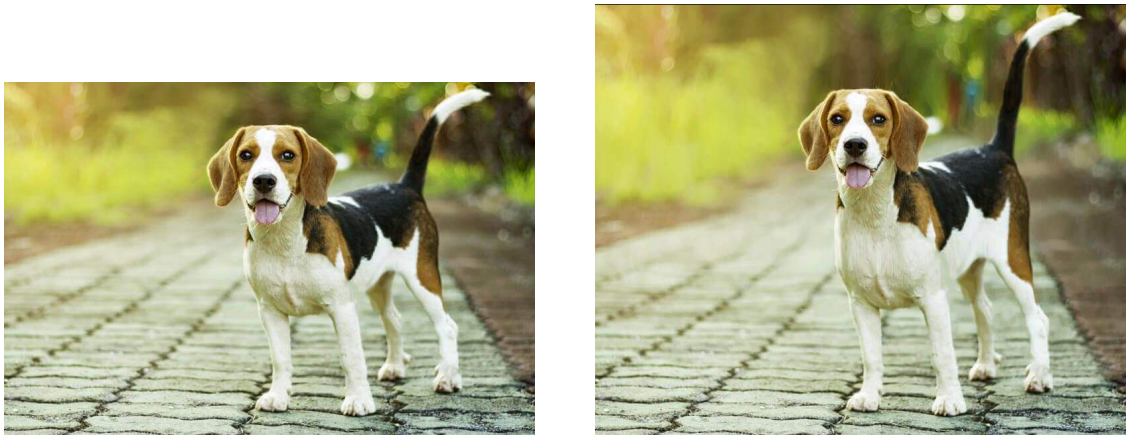

Figure 6: Left Image HxW: 533x800, Right Image HxW: 650x900

Figure 7: Left Image HxW: 400x600, Right Image HxW: 500x800

4.4 Trying to recreate original Image from Scaled up image

The image in the top left is the original image and the image in the top right is the scaled up version of that image. The bottom image was recreated as original image by providing original width and height of the image to the scaled up version of the image.

Figure 8: Recreating original image of a sunflower

Figure 9: Recreating original image of a dog

This image is similarly recovered as described above for the sunflower image.

4.5 Distorted results

Figure 10: Resulting Image to the right is distorted because information cannot be maintained properly

Figure 11: Grass, Rocks appear distorted in right image because important information is regenerated