Size Reduction and Object Removal with Seam Carving and Mask R-CNN

Akul Joshi
University of Illinois
Champaign-Urbana IL

Two fundamental applications of computer vision are the ideas of image size reduction and removal of objects from images; although there exist methods that perform both methods separately, combining these tools into a single algorithm tends to limit the algorithms that we may use. Thus, we re-introduce the idea of seam carving, proposed first by Shai Avidan and Ariel Shamir, which allows us to find low-energy vertical and horizontal seams within an image that may be removed to reduce the dimensions of the image. We combine this method with the idea of image segmentation using the Mask R-CNN algorithm from the PyTorch library, an idea proposed by Jeremy Day and Noel Raley. In this paper, we combine the level of flexibility of seam carving presented in the former paper with the precision of image segmentation using Mask R-CNN in the second paper.

We start with an explanation of the seam carving method presented in the Avidan and Shamir paper. As a quick aside, we have chosen to implement our seam carving method in Python, as it allows for easy manipulation of images; however, Python slows the actual runtime of the method. In their paper, Avidan and Shamir define an energy function

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

that computes the energies of the image at each stage, after which they compute the vertical and horizontal seams with the lowest energies. In this paper, we have chosen to implement a more specific energy function proposed by Northeastern University which uses filters that serve as vertical and horizontal edge detectors. The filters are specified as

$$\text{Horizontal: } H(p) = \begin{array}{|c|c|c|}\hline 1 & 0 & -1 \\ \hline 2 & 0 & -2 \\ \hline 1 & 0 & -1 \\ \hline\end{array} \qquad \text{Vertical: } V(p) = \begin{array}{|c|c|c|}\hline 1 & 2 & 1 \\ \hline 0 & 0 & 0 \\ \hline -1 & -2 & -1 \\ \hline\end{array}$$

and we compute the total energy of a pixel as

$$E(p) = \sqrt{H(p))^2 + (V(p))^2}.$$

Then, to compute the seam to remove from the image, we randomly generated a number between 0 and 1. If this number is less than 0.5, we create a vertical seam; otherwise, we create a horizontal seam by transposing the image, finding a vertical seam, and then re-transposing to recover the original image. The seam finding method itself simply starts at the top of the image and iterates through the pixels; for each pixel (save for those in the first row), the algorithm finds the minimum energy between the top three neighbors, stores the corresponding pixel as the parent of the current pixel, and records the total cumulative energy of the parent pixel summed with the precomputed energy of the current pixel as the total cumulative energy of the current pixel. Then, after the method computes the total cumulatives energies of the final row in the image, it computes the column index corresponding to the minimum total cumulative energy and backtracks through the rows using the stored parent pixels to find the seam.

We now delve into the image segmentation portion of our algorithm, specifically the Mask R-CNN idea presented in the Day and Raley paper. The Mask R-CNN architecture is not something that has to be recreated by us; through the PyTorch library, we are granted access to a trainable Mask R-CNN which can be used to compute object masks on images external to training. Thus, our method works as follows: given an image from which we wish to segment an object, we specify a bounding box that serves as the boundary of our image patch. This image patch is then taken and converted into a PyTorch tensor that may be passed into our model predictor, which returns a dictionary containing a tensor of masks, each the size of the image patch, that the user may peruse. From this list, the user may select the indices of masks that they wish to use in their carving; the algorithm then takes these and creates a composited mask, expands the size of the mask to be the same as the input image, and subtracts this mask times some constant from the energies computed during the seam carving process, allowing us to classify those pixels that we wish to remove as low energy.

Using this proposed seam carving method, as well as the proposed image segmentation method of utilizing a Mask R-CNN, we see very good results, as shown in the "Results" portion of this paper. When only applying the seam carving algorithm without the use of a defined mask, we see that the pixels and seams which are the lowest in energy are removed towards the beginning of the process, whereas pixels and seams which are the highest in energy tend to remain present within the image until the size of the image is so small that they must be removed. This application of our algorithm allows us to compress our images to a smaller size for more effective file transfer or reduce the size of images for display on smaller screens without losing much of the important information present within the image. Furthermore, when applying the image segmentation methodology using Mask R-CNN, we see that the objects that we have bounded in the image and computed masks for are removed very well. When we solely derive masks and attempt to use all of these masks to compute a composite mask, we may get some noise in terms of the pixels which are meant to be removed from the image; however, although it is more work, allowing the users to specify which of the created masks to utilize when computing the composite mask allows for some level of denoising. We may, in essence, choose the masks which best correspond to the object that we want removed from the image, and through this, we may more effectively allow our seam carving algorithm to focus on the removal of those objects.

Although the algorithm itself works fairly well, translating it from the paper and writing it in Python was not without its challenges, many of which could definitely be improved upon.

The original seam carving paper presented by Avidan and Shamir includes a generalized energy computation as well as a specific energy function in the form of histogram of gradients; however, the former was too general for my application whereas the latter was too complex and unnecessarily computationally heavy; thus, it was vital that we found an energy function which served as a good balance between the aforementioned, and the edge-detection-based filter method presented by Northeastern University served as a good compromise. Another challenge was actually choosing which programming language to choose. As mentioned previously, Python tends to be a very good language for image manipulation through the use of numpy arrays; however, the actual running time of the seam carving and image segmentation algorithm using Python tends to be many orders of magnitude slower than the same algorithm in a language such as Java or C++. Thus, choosing between programming languages was vital going into the composition of the algorithm; ultimately, we decided on Python as our language, as the running time of our algorithm mattered less than the ease of writing our method, especially when we take into account that we are writing the first draft of our algorithm. One final challenge was the implementation of masking capability through Mask R-CNN, not specifically computing the masks but rather finding a suitable composite. The paper by Day and Raley is vague in terms of how they choose to use and, thus, we needed to make a design decision that would impact how the algorithm would function not only at a user-level but also in terms of its effectiveness. Ultimately, we decided to display all of the masks resulting from the predictions of the Mask R-CNN architecture and allow for our users to decide which masks to use to create a final composite mask; this not only results in an effective result from the image segmentation process but also allows the user to have a greater say in the mask that they use to segment objects from their main input image. In spite of these challenges, the seam carving and image segmentation algorithm that we have created has resulted in a very effective result and I believe that our combination of features from both the Avidan and Shamir paper and the Day and Raley paper warrants full points in terms of the challenge/innovation portion of grading.

The seam carving and image segmentation method that we have presented here is not limited in scope to the functionality that we have provided by any means. Although our implementation allows us to specify a number of iterations for seam carving (or until the complete removal of an object from an image), we may also further allow our algorithm to reduce the image down to a certain size simply by creating a new version of the seam carving method that runs until the picture reaches that size. Another potential extension of our application could be the introduction of another energy function, such as the histogram of gradients (HoG); although our application functions well without its use, HoG is a more effective energy function to utilize for size reduction without information loss, even if it is more complex than the energy function we have presented above. Finally, we have focused primarily on seam carving, or the removal of seams to reduce the size of an input image; however, Avidan and Shamir have also proposed a method to increase the size of an image using seam insertion, which tends to be a more complex process than seam carving but allows for us to not only optimize images for smaller screens but larger screens as well.

Overall, we have created a method for seam carving and image segmentation that takes an input image and decreases the size of that image while also allowing us to specify an object to remove through the use of the Mask R-CNN architecture. As we have shown from our results and our analysis, the algorithm we have presented functions very well for our application, not only giving us a smaller image in an optimized manner but giving the user the ability to choose which objects they deem important and unimportant.

Results

Seam Carving

Original Image

Image after 200 Iterations of Seam Carving

Seam Carving with Mask R-CNN

Original Image with Bounding Box

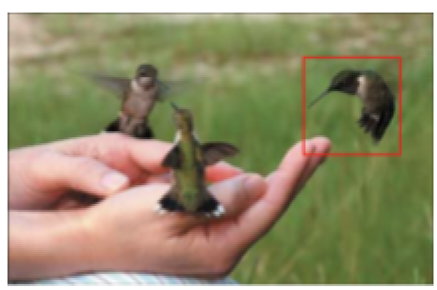
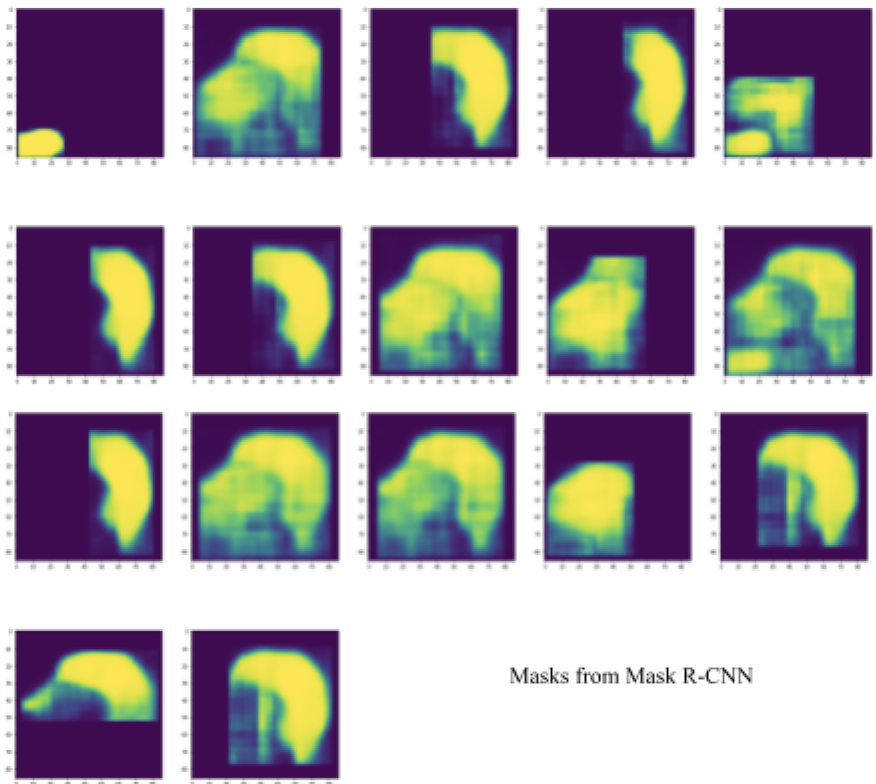


Image Patch



Masks from Mask R-CNN

Composite Mask

Expanded Size Mask

Image with Object Removed

References

Shai Avidan, Ariel Shamir. Seam Carving for Content-Aware Image Resizing. https://perso.crans.org/frenoy/matlab2012/seamcarving.pdf

Jeremy Day, Noel Raley. Combining Image Segmentation and Seam Carving for Automated Object Removal. https://jaday.io/assets/blog/combining-image-segmentation-and-seam-carving-for-automated-object-removal/combining-image-segmentation-and-seam-carving-for-automated-object-removal.pdf

Northeastern University. Assignment 9: Squeezing Pictures. https://course.ccs.neu.edu/cs2510a/assignment9.html

Google Drive Link. https://drive.google.com/drive/folders/1ZiE5CoM6V_nEiOY4C3KFoQ3fjrs3SjSw?usp=sharing