CS269 Project Report:
Automatic Portrait Segmentation and Synthetic Depth-of-Field

Davis Cho
chodavis@g.ucla.edu
005219569

Mike Sun
michaelmusun@ucla.edu
505228712

Yinxue(Yolanda) Xiao
yolandaxiao7@gmail.com
904581627

Abstract

We present an image processing pipeline that automatically adds the Bokeh effect to human-portrait images. Our core focus is on the step of developing a robust segmentation method using traditional methods. We apply three separate image segmentation approaches including 1) active contour models with CNN, 2) level-set active contours, and 3) graph-cut. We perform a comparison across all three methods. A Gaussian filter is applied on the background to produce the final image with synthetic depth-of-field. Our experiments show that the output images are produced with decent bokeh effects. They also show that graph-cut achieves the best segmentation accuracy and final result compared to the other two.

1 Introduction

Smartphones are one of the most ubiquitous technologies in the world. Within the smartphone industry, cameras are one of the biggest factors consumers care about. High quality photography has become very accessible to the general population, as almost anyone today can take a high resolution picture with the phone in their pocket. Being able to achieve the same photo effects of a DSLR with a mobile phone camera is the next logical goal, and one thing that can be automated is synthetic depth-of-field. DSLRs produce a nice Bokeh effect when the big aperture is used, which blurs the background nicely. It is thus desirable for mobile cameras to achieve the same result. However, it is almost impossible for mobile cameras to obtain background blur from the optical process through their small apertures and short focal lengths. We thus explore techniques in computational photography to produce synthetic depth-of-field.

The process of producing the Bokeh effect is divided into two steps: image segmentation and filtering. Most recent works in image segmentation use deep learning, specifically some variations of Fully Convolutional Networks (FCN). Before the emergence of neural networks, researchers used a wide variety of traditional methods based on physics and mathematical models. One such approach is the snake model. It obtains object boundaries through energy minimization based on the image, elastic beam, and constraints. The level-set active contours model is a variation of the original snake model. It uses interior and exterior area values to perform energy minimization to obtain a closed-form edge[3]. Another approach is graph-cut, which is also based on energy minimization by computing min-cut/max-flow of an image graph [2].

While current methods in deep learning yields high accuracy in image segmentation, this work aims to investigate the traditional methods based on energy minimization. This project applies three different approaches in image segmentation: 1) Active Contour Models with Neural Networks[8], 2) Level-Set Active Contours, and 3) Graph-Cut, on portrait images and compares the results. Note that the snake model approach is combined with a CNN in order to find the best parameters for the snake to achieve automatic segmentation.

In the second part, Gaussian blur is applied on the background image in order to complete the synthetic depth-of-field effect.

2 Related Work

One previous work for the synthetic depth-of-field task on mobile phone camera is the paper from Wadhwa et al.[13]. They use a person segmentation network to obtain a person's mask, and dual-pixel data to compute depth map and disparity. This information is combined to render the final image with a bokeh effect. Since they use neural networks, specifically a variation of U-Net[10], for the portrait segmentation, it doesn't align with the goal of our project. We thus looked at related works in the following three directions.

2.1 Active Contours with Convolutional Neural Networks

Kass et al. first proposed the concept of active contours[6], or snakes, which are splines that climb and conform to the minimum energy in an image. Rupprechet integrated ACM with CNNs that control contour evolution, but the whole boundary detection mechanism was still interactive. A more automatic and less interactive process was proposed by Marcos et al. using Deep Structured Active Contours (DSAC)[9]. This pipeline uses a CNN to learn and output optimal parameters for the ACM energy function to use in order to achieve maximal IoU from the segmentation that is determined. The IoU is then fed back into the CNN to determine the appropriateness of the parameters. Thus, certain parameters like the membrane term, thin plate term, and balloon term are automatically determined by the CNN. Hatamizadeh et al. extended the CNN automation of the ACM with the proposed Deep Convolutional Active Contours (DCAC)[5]. DCAC can be considered an extension for the DSAC and actually fulfills the fully end-to-end training criteria by incorporating all parameters and initializations into the training, whereas compared to the DSAC, many portions that were assumed to be given, such as the intialization of the ACM contours.

2.2 Level-Set Active Contours

An alternative method to segmenting the individual out of the portrait is the well known level set contouring method, also known as Chan-Vese method[3], and has been used in various papers as the baseline algorithm to modify to achieve better results[7]. The driving motivation behind why this method was chosen as an alternative lies in the method's disregard to edges, which can serve as a great advantage. The energy function for the Chan-Vese method compares an energy within a segmentation and compares it with the energy outside of its segmentation and attempts to learn a global minimum energy. This method, once again, does not rely on the edges of the image, which can advantageous especially when it comes to segmenting unusual or asymmetric shapes because it no longer has the smoothness or length limitations that the normal ACM model has.

2.3 Graph-Cut and Superpixels

Previous works have also approached image segmentation through graph-based methods. Grieg et al.[4] first formulated that the energy of an image can be seen as a sum of the image data and the interaction potential between neighboring pixels. A later work by Boykov and Kolmogorov introduced an efficient min-cut/max flow algorithm for energy minimization in vision, which can be used for image segmentation. In general, graph-based methods first constructs a graph based on a image, creating nodes from image pixels or features. It selects two nodes as a source and a terminal, and assigns costs to all the edges in the graph. A cut partitions the nodes into two disjoint sets, with source node and terminal node in each set. The goal is to find a cut with the minimum sum of costs of edges among all cuts. In image segmentation, the cut can be viewed as the boundary that separates an image into two parts with foreground and background.

Using every single pixel in an image for the graph construction stage is every inefficient. It thus leads to creating images with low-level atomic regions for graph construction, and superpixels is a common approach. Superpixel algorithms group pixels in an image based on their similarities into sub regions. It divides an image into smaller parts based on its low-level features. Superpixels approaches include graph-based algorithms such as normalized cuts and graident-ascent-based algorithms such as watershed. Simple Linear Iterative Clustering(SLIC) is a superpixels method proposed by Achanta et al.[1] with state-of-the-art boundary adherence and good performance. It generates superpixels by performing k-means to cluster pixels based on color similarity and proximity in a

5-dimensional space.

3 Segmentation Approaches

3.1 Active Contours with Convolutional Neural Networks

Our first approach uses a convolutional neural network (CNN) that learns to adjust an ACM, based on the DSAC architecture. In training, it adjusts the parameters of an ACM on an image so that the ACM best fits to a provided ground-truth mask. The CNN learns the energy function that allows the ACM to form a closest-matching polygon. The parameters are:

α - membrane term - length penalization.

β - thin plate term - curvature penalization.

κ - balloon term.

We tested various approaches in adapting and fine-tuning the network architecture. We altered the size and number of layers, as well as changed the shape of the initialization. However, these showed no noticeable change in the results. An approach that yielded some effect was adjusting the initial biases of each of the three learned parameters. Testing showed that changing them changed the shape of the resulting contour, although performance was not enhanced.

The resulting segmentation from our generated contours is evaluated against the original manually-segmented image to compute the IoU (Intersection over Union) loss. This loss is used both in training and evaluation of our models.

3.2 Level-Set Active Contours

Our second approach was to use the level-set active contour method. This approach, to reiterate, does not rely on edges, and instead uses interior area values versus exterior area values to iteratively determine what would provide the minimal energy. In this approach to segment the portrait, we used tested for and selected suitable parameter values via trials of different combinations to use in a basic level-set segmentation method implemented in the Sci-Kit Image library. The parameters primarily explored for the level set method are:

μ - value between 0 and 1, where 1 is used for most ill-defined shapes

$\lambda 1$ and $\lambda 2$ - two lambda values. They should generally be the same unless the background is very different from the foreground

After trying multiple value combinations of μ and λ, it was determined that a μ value of 0.5 and $\lambda 1$ and $\lambda 2$ values of 1 were most appropriate, since humans are somewhat ill-defined despite being generally symmetric along the y-axis. The λ values are both 1 because most of the images actually had the foreground color similar to that of the background color. We must emphasize "most appropriate" λ and μ because although these values did not produce the best IoU, they output filters which are clearly better. For example, the combination of μ=0.25, $\lambda 1$=0, and $\lambda 2$=1 actually gave the best IoU score. However, the filters this combination outputted were often using the whole image as one segment, or on the other extreme, not segmenting any part of the image at all. Thus, we manually analyzed which produced the best outputs among a select few parameters. Details and examples can be seen in Figure 3.

The resulting segmentations from the level-set approach is also evaluated against the original hand-segmented ground truth images via the IoU criteria.

3.3 Graph-Cut with SLIC Superpixels

This approach first runs SLIC to obtain the superpixels of an image to construct a graph, then performs graph-cut to obtain the image segmentation.

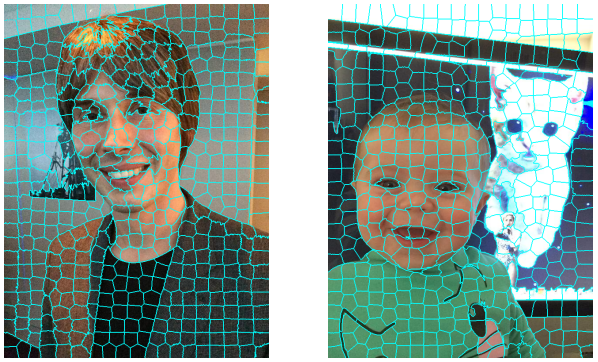
3.3.1 SLIC Superpixels

Since it is difficult for the active contour model to find exact boundaries of human figures due to the complicated image features, we aim to find a traditional method that emphasizes edges and boundaries. The boundary adherence attribute from superpixels thus comes to our attention.

Superpixel algorithms group similar pixels together into atomic regions. SLIC, in particular, uses k-means to cluster pixels together that are of similar color and proximity, and it exhibits good boundary adherence as well as efficiency. The boundary adherence capability makes

sure that all the important edges are preserved. Image segmentation can be done if the correct set of superpixels are chosen and connected.

We run the SLIC method from the scikit-image implementation of the original approach. Two sample results are shown in Fig.1. It shows the boundary adherence result nicely, as the superpixels keep the human figure boundaries intact. The task now becomes how to group superpixels into two sets so that the foreground and background belongs to each one respectively.



(a) Example 1 (b) Example 2

Figure 1: SLIC Superpixels Example

3.3.2 Graph-Cut

The graph-cut methods finds a minimum cut of a graph to divide it into two disjoint sets. It thus satisfies our need for image segmentation, especially portrait segmentation since it separates one image into two parts: a foreground person and a background view.

The goal is to first construct a graph based on an image and then run graph-cut on it to obtain the segmentation. Similar to Shilkrot's implementation of graph-cut[12], we construct the graph of an image based on the color histograms of the superpixel segments. Given the SLIC superpixels, it computes the center and color histogram of those segments, as well as the neighbors of the segment centers. These values are used to set the weights of the edges in the graph. Many current graph-cut applications are interactive and require user input to define a source node and a terminal node. However, since the goal of this project to automate segmentation and filtering, we define the source and terminal nodes programmatically. Note that the dataset is preprocessed into images of size 600x800, with the figure centered in the middle. We set the center of the image as the source and the top-left and top-right corner as the terminal, given the characteristics of the dataset. PyMaxflow, a python implementation of Boykov and Kolmogorov's graph-cut implementation, is used to compute the min-cut of the constructed graph.

4 Synthetic Depth-of-Field

Given the results of segmentation masks, a wide variety of filters can be applied to the background image to achieve different effects. This project specifically focuses on synthetic depth-of-field, or the bokeh effect.

Bokeh refers to an image in which the background is out of focus. It gives the background a blurred effect so the main subject in the foreground stands out. We aim to produce similar effects using Gaussian blur. Note that Gaussian filter produces blurring quite differently from that of bokeh: it essentially smooths a region, killing the noise, and it not dependent on depth. However, the blurring result can look comparatively similar.

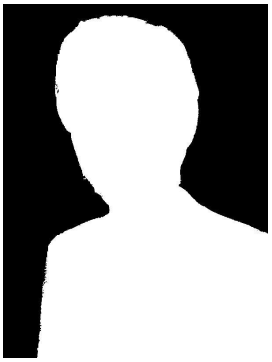
In this project, we first apply Gaussian blur on the original image. The background portion of the blurred image is then concatenated with the foreground portion of the original image to obtain the final result. The segmentation mask produced in the previous section is used to obtain the split of foreground and background image segments. Note that it is important to apply Gaussian blur on the entire image first before it is segmented in order to avoid undesirable artifacts on the segment edges.

5 Dataset

The three different image segmentation approaches are evaluated on the dataset provided in Shen et al.'s paper[11]. They collected 1800 portrait images from Flickr and labeled them with foreground/background segmentation using the quick selection tool in Photoshop. The images were run through a face detector that scaled and cropped them to a size of 600x800. One such image along with its ground truth mask from the dataset is shown

in Fig.2 as an example.

(a) Input Image



(b) Ground Truth Image Mask

Figure 2: Sample Image and Ground Truth Mask from the Dataset

Fig.3 is an example of how increasing the length penalization parameter appropriately reduces the overall length of the contour. However, the total encapsulated area of the person does not change. Table 2 shows the averaged results of adjusting these parameters in various combinations, yielding little significant results.

Figure 3: Effect of length penalization initialization on contour

6 Results and Comparison

6.1 Segmentation

The average IoU results for all three approaches are shown in Table 1 below. The highest accuracy is achieved using Graph-Cut with a 76.15% average IoU, while the active contours with CNN method has 48.0% and the level-set approach with 51.74%.

	ACM CNN	Level-Set	Graph-Cut
Average IoU	48.0%	51.74%	76.15%

Table 1: Average IoU of the three methods on the dataset

6.1.1 Active Contours with CNN

The results showed that the active contours with CNN approach was unable to produce precise contours under the portrait dataset. Adjusting initial biases for energy, length penalization, curvature penalization, and the balloon term affected results but did not result in particularly proper training or improved IoU loss.

α	β	κ	Mean IOU
0.1	0.1	0.1	0.44
0.01	0.1	0.1	0.45
0.1	0.01	0.1	0.47
0.1	0.1	0.01	0.43
0.99	0.1	0.1	0.44
0.1	0.99	0.1	0.44
0.1	0.1	0.99	0.46
0.01	0.01	0.1	0.48
0.2	0.2	0.8	0.45

Table 2: Mean IoU obtained from adjusting ACM parameters

There are possible explanations to the poor performance. The shape of a human is much more complex than that of a house, which is generally rectangular or at least geometric. The model may have difficulty forming through bottlenecks on the human figure like the neck, and expanding back into the torso.

A question remained of whether the poor results stemmed from weakness in the CNN not adjusting parameters well enough, or in the ACM naturally being a bad model for this dataset. We tested a pure ACM model without the CNN, the results of which supported the lat-

ter. An ACM does well in conforming to just a face, but is unable to conform well to a human's entire upper body, even when manually tuned. The snake encounters too many obstructions to be able to differentiate noise from the precise outline of the human. With the unconstrained and widely varied real-world dataset of consumers taking photos in any location, lighting, angle, and environment, this model would certainly have a difficult time.

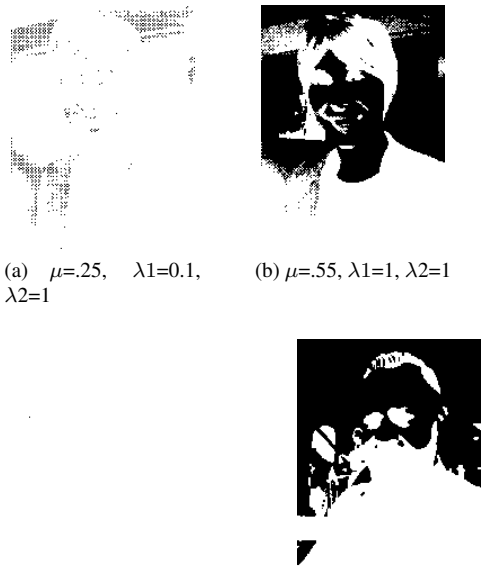
6.1.2 Level-Set

The simple level-set approach attained an average IoU percentage of 51.74% and sample results can be seen in Figure 6. Overall, some images performed relatively well upon inspection, but others did not perform nearly as well because IoU values ranged from 0.2 to 0.9 per image.

μ	$\lambda 1$	$\lambda 2$	Subset Mean IoU
0.00	0.1	0.1	0.402
0.00	0.1	1	0.534
0.00	1	0.1	0.159
0.00	1	1	0.402
0.25	0.1	0.1	0.473
0.25	0.1	1	0.599
0.25	1	0.1	0.005
0.25	1	1	0.464
0.50	0.1	0.1	0.461
0.50	0.1	1	0.591
0.50	1	0.1	0.032
0.50	1	1	0.467
0.75	0.1	0.1	0.446
0.75	0.1	1	0.582
0.75	1	0.1	0.523
0.75	1	1	0.469
1.00	0.1	0.1	0.430
1.00	0.1	1	0.576
1.00	1	0.1	0.067
1.00	1	1	0.471

Table 3: Mean IoU obtained from adjusting Level Set parameters

We had initially hypothesized that learning the parameter inputs into the Chan-Vese segmentation algorithm per image might be able to increase the performance of the



(a) μ=.25, $\lambda 1$=0.1, $\lambda 2$=1

(b) μ=.55, $\lambda 1$=1, $\lambda 2$=1

(c) μ=.25, $\lambda 1$=0.1, $\lambda 2$=1 (Whole image was used as a single segment)

(d) μ=.55, $\lambda 1$=1, $\lambda 2$=1

Figure 4: Level-Set Parameter Comparison

algorithm (this is why we proposed using a CNN to learn the parameters to use for the energy function of the level-set method). However, upon further testing and analysis, it was quickly realized that this was not necessarily true. Learning the best parameters which give us the highest IoU does not always give us the "best" results in that the filter outputted does not conform like one would expect. This could be because IoU may not be the best judging criteria for level set when it comes to learning the parameter values.

Additionally, it was discovered that the true bottleneck keeping the IoU score is low in the final segmentation of the portrait is the final segmentation always tended to include many more objects in its segmentation than necessary. When only the segments that correspond to the person are included in the final segmentation, the improvements made by changing parameter values are much less significant in comparison.

Methods on how to remedy this are mentioned in future

work, but this problem is an enormous contributing factor as to what caused us to explore the Graph-Cut method.

6.1.3 Graph-Cut

Examples of the segmentation mask results from the graph-cut approach is shown in the second row of Fig.7. It generally captures the rough shape of the human figure well. However, the details of the edges are still relatively coarse. Since the graph is constructed based on color histograms of the superpixel segments, it is not as robust in the case when foreground and background have similar colors. An example can be seen in the image in the 5th column of Fig.7, where the hair is blended into the background.

6.1.4 Comparison

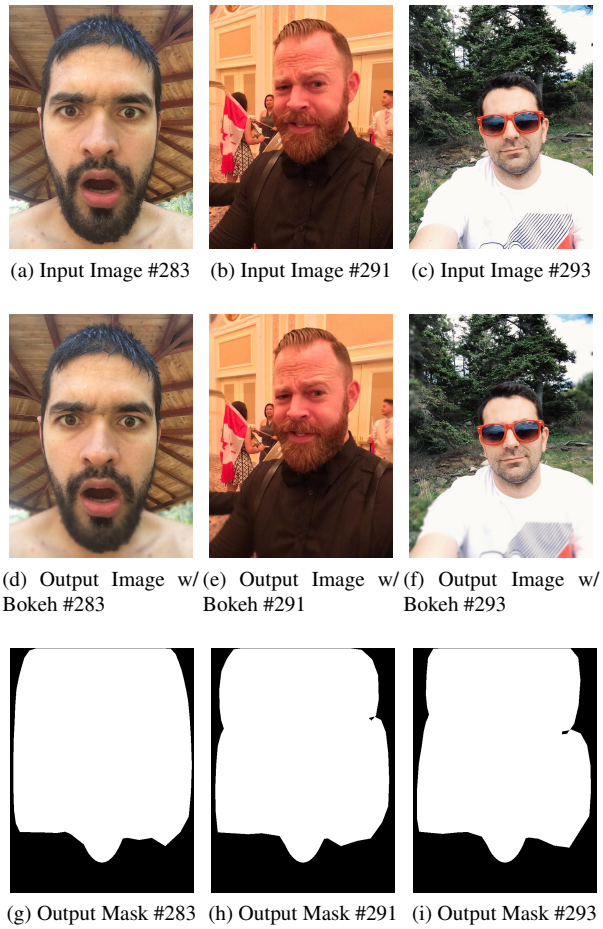
Graph-cut performs the best out of the three image segmentation approaches we investigated while the DSAC model performed the worst.

This is likely because the graph-cut algorithm seeks to split the photo into its two most different parts. This naturally fits our problem statement of segmenting out the foreground and background of an image. The graph-cut's use of superpixels could also contribute to its good performance, since it captures low-level image features such as the boundaries well.

Snakes, on the contrary, are limited by their property of being one continuous entity. While the graph-cut can make sharp turns along the dynamic border of the person in an image, each part of the snake has a certain dependence on the overall shape of the snake. For example, when the snake tries expanding to fill the wide shoulders, the narrow parts at the neck are pulled to expand as well.

Level-set segmentations are not bound by the same limitations of snakes in that it needs to be one continuous entity which relies on edges, but still suffers in its own rights in that it lacks a formal method to identify which segmentation is that of interest to us. It would require some form of prior, in addition to training the input parameters, in order to determine what the area of interest is, otherwise it would simply use every segment in the image as one large segmentation, which would cause the IoU value to suffer drastically.

6.2 Filtering



(a) Input Image #283 (b) Input Image #291 (c) Input Image #293

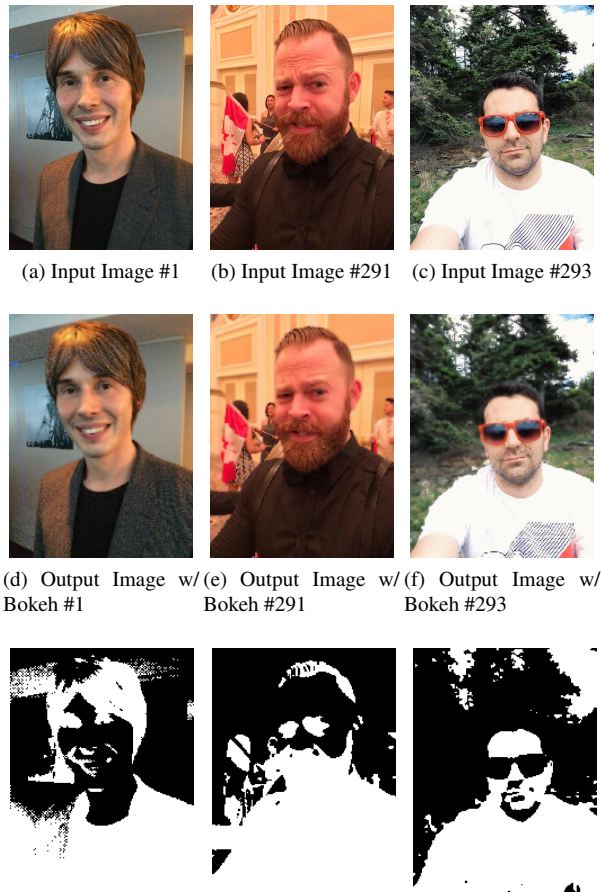
(d) Output Image w/ Bokeh #283 (e) Output Image w/ Bokeh #291 (f) Output Image w/ Bokeh #293

(g) Output Mask #283 (h) Output Mask #291 (i) Output Mask #293

Figure 5: Image Result Comparison for CNN+ACM Segmentation

The CNN+ACM approach's poor performance lends itself to a poor yet consistent mask, which does not accomplish the bokeh effect. The ACM does not conform to individual cases well at all, so the same parts of every image get blurred. The level-set method suffers similar disadvantages, but does perform slightly better than the CNN+ACM method, although differences between the two are mostly unnoticeable to the naked eye.

The output images with synthetic depth-of-field from



(a) Input Image #1 (b) Input Image #291 (c) Input Image #293

(d) Output Image w/ Bokeh #1 (e) Output Image w/ Bokeh #291 (f) Output Image w/ Bokeh #293

(g) Output Mask #1 (h) Output Mask #291 (i) Output Mask #293

Figure 6: Image Result Comparison for Level Set Segmentation

graph-cut are shown in the third row of Fig.7. The images on the left 4 columns produce decent results with bokeh effect: the background is blurred, and the person is emphasized as the main subject.

The final result is greatly dependent on the image segmentation output. We observe that the general outline of segmentation is essential to producing good final results. However, the details next to the segmentation line are less emphasized. As shown in the images in the first 4 columns, even though the segmentation masks are not 100% correct, the final result is decent.

There are still failure examples, as shown in the last two columns. In the picture on the 5th column, the girl's forehead is incorrectly identified as part of the background and is consequently blurred out. In the picture on the 6th column, the background is falsely identified as foreground, thus not resulting in any blurring.

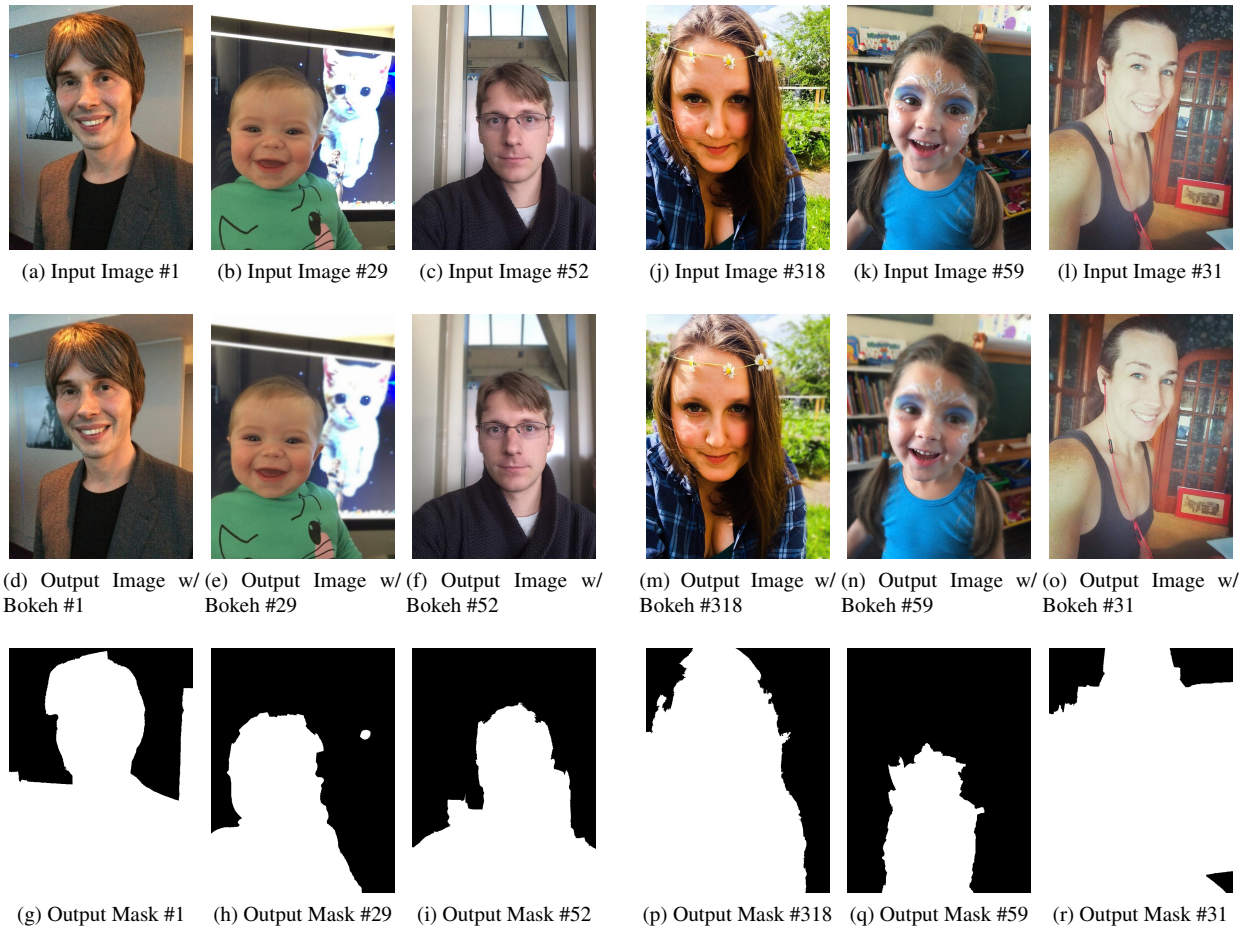
7 Conclusion and Future Work

We achieved synthetic depth-of-field through a combination of portrait segmentation and background filtering. In the image segmentation step, we compared three different methods including active contour models with CNN, a level-set approach, and graph-cut and showed that the graph-cut approach is most suitable to the task of portrait segmentation, relative to the active contour models and the level-set approach. The graph-cut approach was able to remedy a few of the drawbacks that exist in the ACM and level-set methods to attain better results both quantitatively and qualitatively in this particular task of portrait segmentation for the bokeh effect.

In terms of future work, we may be possible to improve performance by exploring and adjusting different parameters for the ACM+CNN approach. There may be more expressive parameters than the ones we explored here that would lead to more accurate segmentation. For example we could extend DSAC to incorporate a more comprehensive set of parameters, such as the geometric shape initialization.

We can see from the outputted segmentations in Figure 6 that the Chan-Vese method actually segments many things in the portrait, not just the person. One possible way to try and improve the overall performance may be to select one particular segment(or a few segments in the case that the person is split into multiple segments) in the image to be used as the segmentation for the person. This way, we can learn which segment(s) would provide the best IoU. Learning which segments would produce the best IoU can be done via a CNN that has as input all the segments and output binary values which represent booleans on whether or not the corresponding segment belongs in the final segmentation.

For the graph-cut approach, improvements could be done in the graph construction section in order to obtain



(a) Input Image #1 (b) Input Image #29 (c) Input Image #52 (j) Input Image #318 (k) Input Image #59 (l) Input Image #31

(d) Output Image w/ Bokeh #1 (e) Output Image w/ Bokeh #29 (f) Output Image w/ Bokeh #52 (m) Output Image w/ Bokeh #318 (n) Output Image w/ Bokeh #59 (o) Output Image w/ Bokeh #31

(g) Output Mask #1 (h) Output Mask #29 (i) Output Mask #52 (p) Output Mask #318 (q) Output Mask #59 (r) Output Mask #31

Figure 7: Image Result Comparison for Graph-Cut Segmentation

a higher segmentation accuracy. The current graph construction is only based on the color histogram, however, the contour and boundary information could be used to compute the weights for the edges of the graph. Other methods related to superpixel grouping could also be a possibility in increasing the segmentation accuracy. As for the bokeh effect, the final result could potentially benefit from the addition of depth information.

References

[1] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(11):2274–2282, Nov 2012.

[2] Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26:1124–1137, September 2004.

[3] T. Chan and L. Vese. Active contours without edges. *In IEEE Transactions on Image Processing, 2001,*

DOI:10.1109/83.902291, 2001.

[4] D. M. Greig, B. T. Porteous, and A. H. Seheult. Exact maximum a posteriori estimation for binary images. *Journal of the Royal Statistical Society, Series B: Methodological*, 51:271–279, 1989.

[5] A. Hatamizadeh, D. Sengupta, and D. Terzopoulos. End-to-end deep convolutional active contours for image segmentation. *arXiv preprint arXiv:1909.13359*, 2019.

[6] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, 1(4):321–331, Jan 1988.

[7] A. Kristiadi and P. Pranowo. Deep convolutional level set method for image segmentation. *Journal of ICT Research and Applications*, 11:284, 12 2017.

[8] D. Marcos, D. Tuia, B. Kellenberger, L. Zhang, M. Bai, R. Liao, and R. Urtasun. Learning deep structured active contours end-to-end. *CoRR*, abs/1803.06329, 2018.

[9] D. Marcos, D. Tuia, B. Kellenberger, L. Zhang, M. Bai, R. Liao, and R. Urtasun. Learning deep structured active contours end-to-end. 2018.

[10] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. *CoRR*, abs/1505.04597, 2015.

[11] X. Shen, A. Hertzmann, J. Jia, S. Paris, B. L. Price, E. Shechtman, and I. Sachs. Automatic portrait segmentation for image stylization. *Comput. Graph. Forum*, 35:93–102, 2016.

[12] R. Shilkrot. Revisiting graph-cut segmentation with slic and color histograms, 2017.

[13] N. Wadhwa, R. Garg, D. E. Jacobs, B. E. Feldman, N. Kanazawa, R. Carroll, Y. Movshovitz-Attias, J. T. Barron, Y. Pritch, and M. Levoy. Synthetic depth-of-field with a single-camera mobile phone. *CoRR*, abs/1806.04171, 2018.