Effect of stroke width on semi-automatic image segmentation algorithms

Katrina Hoffert under supervision of Dr Mark Eramian

2016-2017

Abstract

Semi-automatic image segmentation involves user interaction to some degree to assist in segmenting an image into multiple parts. Specifically, we looked at the category of algorithms that involve the user marking some areas as foreground and background. Past studies at the University of Saskatchewan had used user strokes or points that were always one pixel wide. This study looked at how the width of the stroke size influenced the results, with respect to accuracy (measured with the Dice similarity coefficient) and reproducibility (measured with the general Tanimoto coefficient).

All user strokes were dilated between 1-4 pixels and then used to segment the image. This was done with the Boykov max flow graph cut algorithm and also tested on the OneCut algorithm. We used images and user annotation data from two past studies, which both used images from the Berkeley Segmentation Dataset 500.

We found that dilating the strokes to have a radius of 4 pixels wide increased the mean DSC by between 0.010 to 0.065 and mean GTC increased between 0.061 and 0.069 ($p < 0.01$ for comparisons between non-dilated and 4 pixel dilation) with the primary segmentation algorithm. The result was also more significant when the user had more time to markup the images. This indicates that thicker strokes tend to get more accurate and more consistent results with thicker stroke values for our segmentation algorithm.

1 Introduction

1.1 An intro to image segmentation

The goal of image segmentation is to partition an image into distinct segments. Semi-automatic image segmentation utilizes some degree of user interaction to assist in this segmentation (but without making the user do all the work). Specifically, our study is concerned with the type of semi-automatic image segmentation that segments a foreground object from its background.

The method of user interactivity that this study looks at is where the user annotates what is foreground and what is background via applying points or strokes to an image. The segmentation program then takes in those user annotations (the “label image”) and the image to be segmented and outputs a binary segmentation mask (which could be used to, say, separate the foreground and background).

Figure 1 demonstrates the types of images involved in the process. At the top left we have the image we want to segment. The top right shows an example of user annotations (red = foreground, green = background). The bottom left shows the resulting segmentation and the bottom right shows the ground truth, which is the correct segmentation.

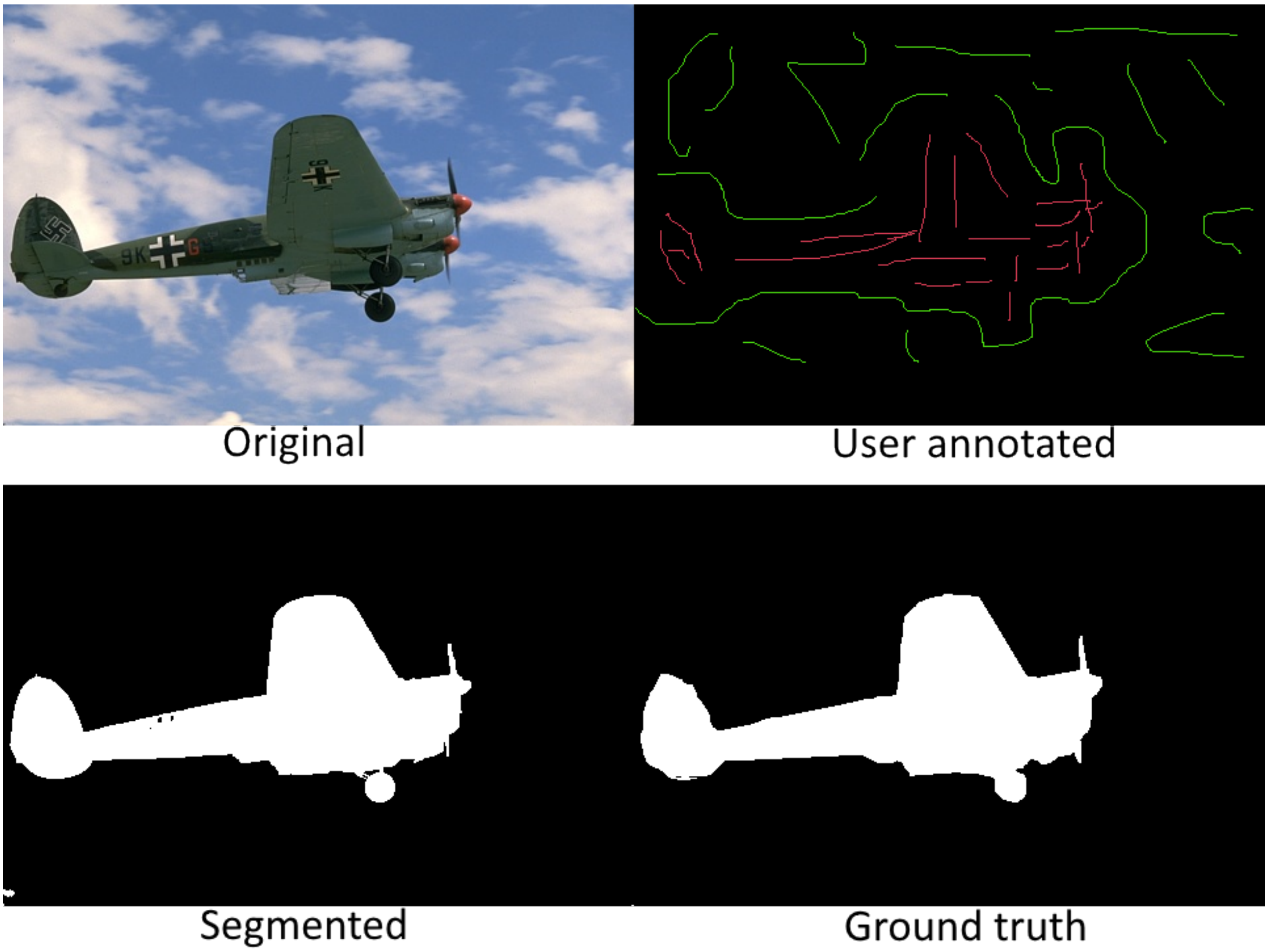

Figure 1: The files involved in segmenting an image.

1.2 Problem

The question that we're trying to solve is "what is the impact of the width of these strokes?" Or in other words, how much does the stroke width affect the quality of image segmentation? All the previous experiments in semi-automatic image segmentation conducted at the University of Saskatchewan had used 1 pixel wide strokes and points. There's the question of how different the segmentation result would be if the strokes were thicker (if at all).

Thicker strokes means that there's more annotations for the algorithm to use as "seeds" for segmenting and that any noise would have less of an impact. However, it's hard to say how much this could impact any given algorithm (and the results are surely algorithm dependent, as well). Thicker strokes can also mean that errors could be introduced due to the strokes now overlapping both the foreground and background. The impact of errors may vary with the algorithm, as well.

1.3 Data sources

This study uses two previous studies to provide user annotations that we can work on:

1. In *Analysis of User Input Methods For Semi-Automatic Segmentation* by Steven Rau [1], the study contrasted the differences between user annotations that were strokes vs points.

2. In *Effects of Time Pressure on Semiautomatic Segmentation* by Yuanxia Li [2], the study contrasted the differences between 15, 25, and 40 second time limits for annotating each image.

Both of these studies provide the needed format of having the forground and background marked by the user. They also do introduce some additional independent variables that we must treat separately (strokes versus points and 15s versus 25s versus 40s time pressures). All data in the study by Li used points.

All of the images in these studies comes from the Berkeley Segmentation Data Set and Benchmarks 500 [3] (BSDS500). Figure 2 provides examples of images that would be segmented. Note how some images are clearly easier to segment than others (eg, because the foreground and background have pronounced differences). The BSDS500 also provides us with the ground truth images that we need to analyze the segmentation results.

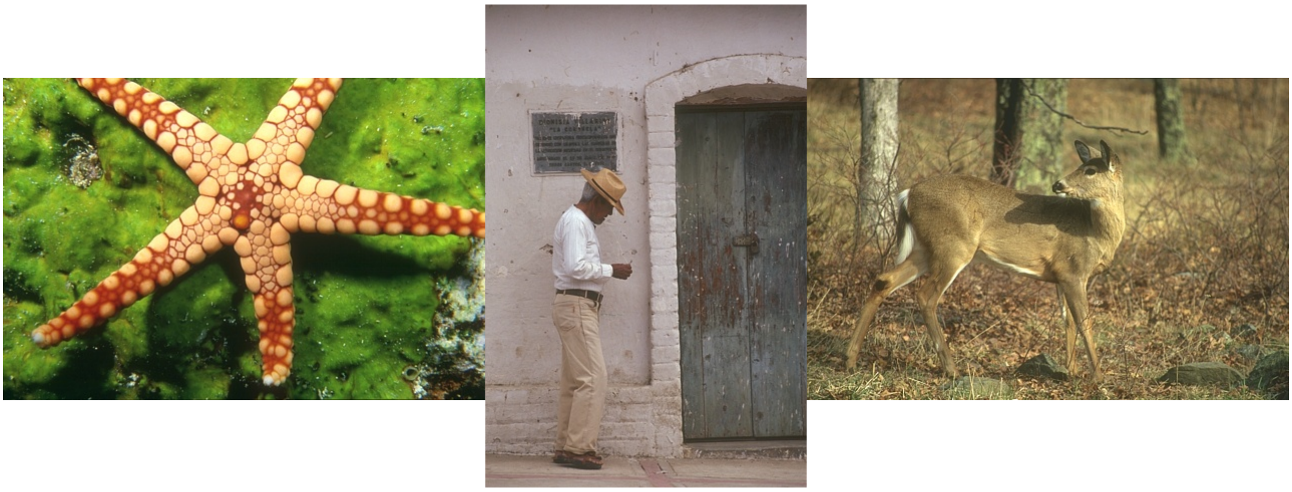

Figure 2: Example files from the BSDS500.

Both of these studies collected data by having a number of users (13 in Rau's study and 19 in Li's study) annotate a variety of images in the manner described above. Users were given instructions on how to mark up the images and given a short training period (that was not counted). Rau's study had users annotating with points and strokes, while Li's study had users segmenting images with different time limits (15, 25, and 40 seconds).

The Rau study provided 650 label images (325 with strokes and 325 with points) and the Li study provided 1425 label images (475 of each time pressure). Adding in the 4 different radii of dilation per image made for a total of 10375 images to segment.

2 Experiment structure

2.1 Overview

The experiment was structured in a pipeline fashion, where all images are processed in a variety of steps, all of them at once. For the most part, all the images from different studies can be handled in the same way (other than ensuring that we preserve study-specific information about them). Then the main steps that we applied to all these images were:

1. Dilate: The foreground and background strokes of the user annotations were dilated by 1-4 pixel radii.
2. Segment: All the label images (dilated and undilated) were then run through segmentation programs to produce binary segmentation masks.
3. Analyze: We computed measures for the images, which were stored in tables.
4. Statistics: We computed overall statistics of these measures.

2.2 Dilation

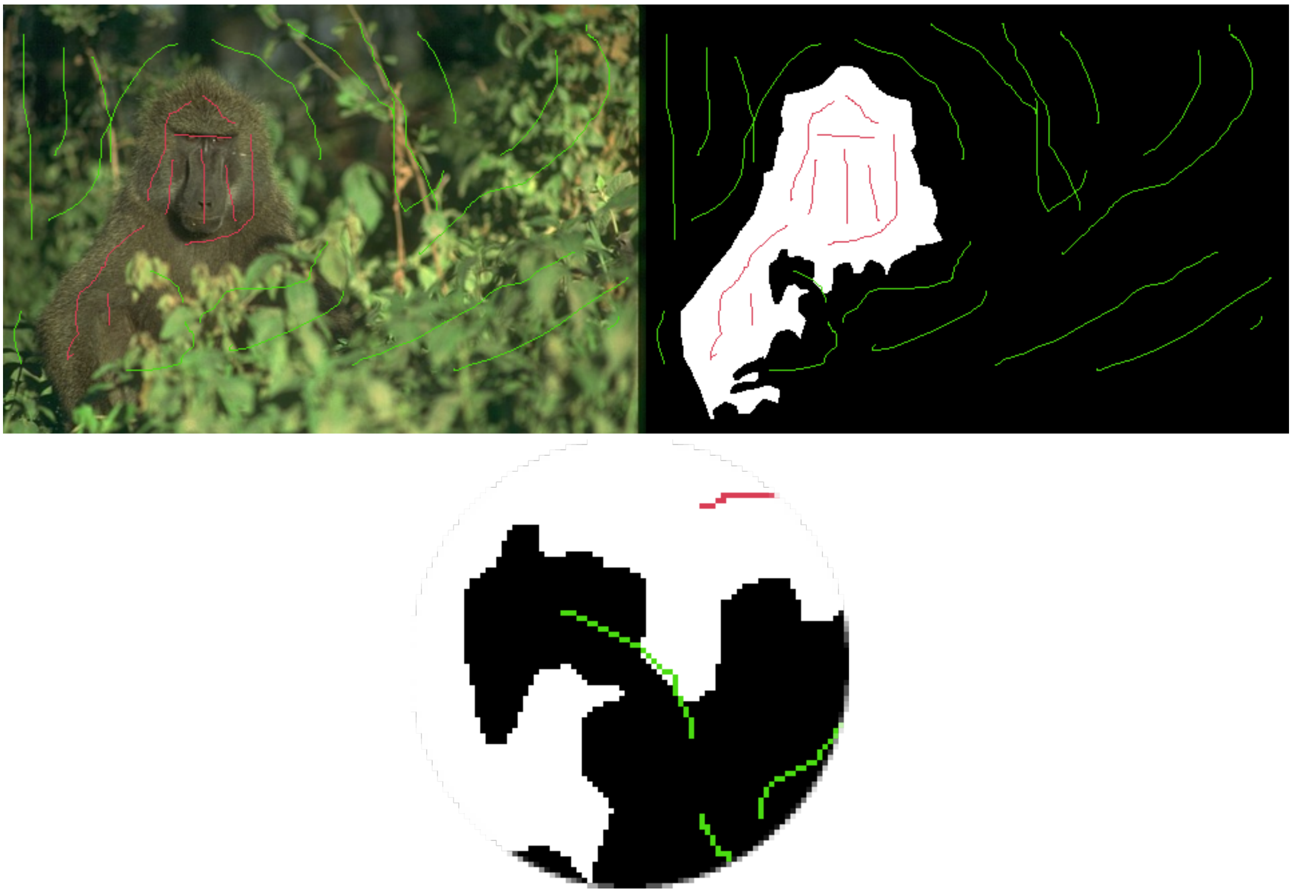

Figure 3: An example of a user error in the annotation process.

The dilation of the label images is the primary independent variable that we're looking at in this study. The dilation was done by setting new pixels as either foreground or background due to their proximity to existing labels. Or put otherwise, all non-zero label pixels would be extended some radius from their center (a value of zero in the label image indicates no label). This dilation is done in a circular fashion, meaning that if the dilation radius is r, then all pixels with a Manhattan distance of less than or equal to r will be given the same label.

The fact that dilation is a radius means that the width of a stroke is $2r + 1$. Hence, our upper limit of radius 4 pixel dilation means that strokes have a width of 9 pixels.

In an attempt to avoid introducing *new* errors into the image that the user perhaps wouldn't have made if they were aware of the width of the stroke, we take an approach that depends on "reading the future" by glancing at the ground truth and not dilating erroneous labels. For

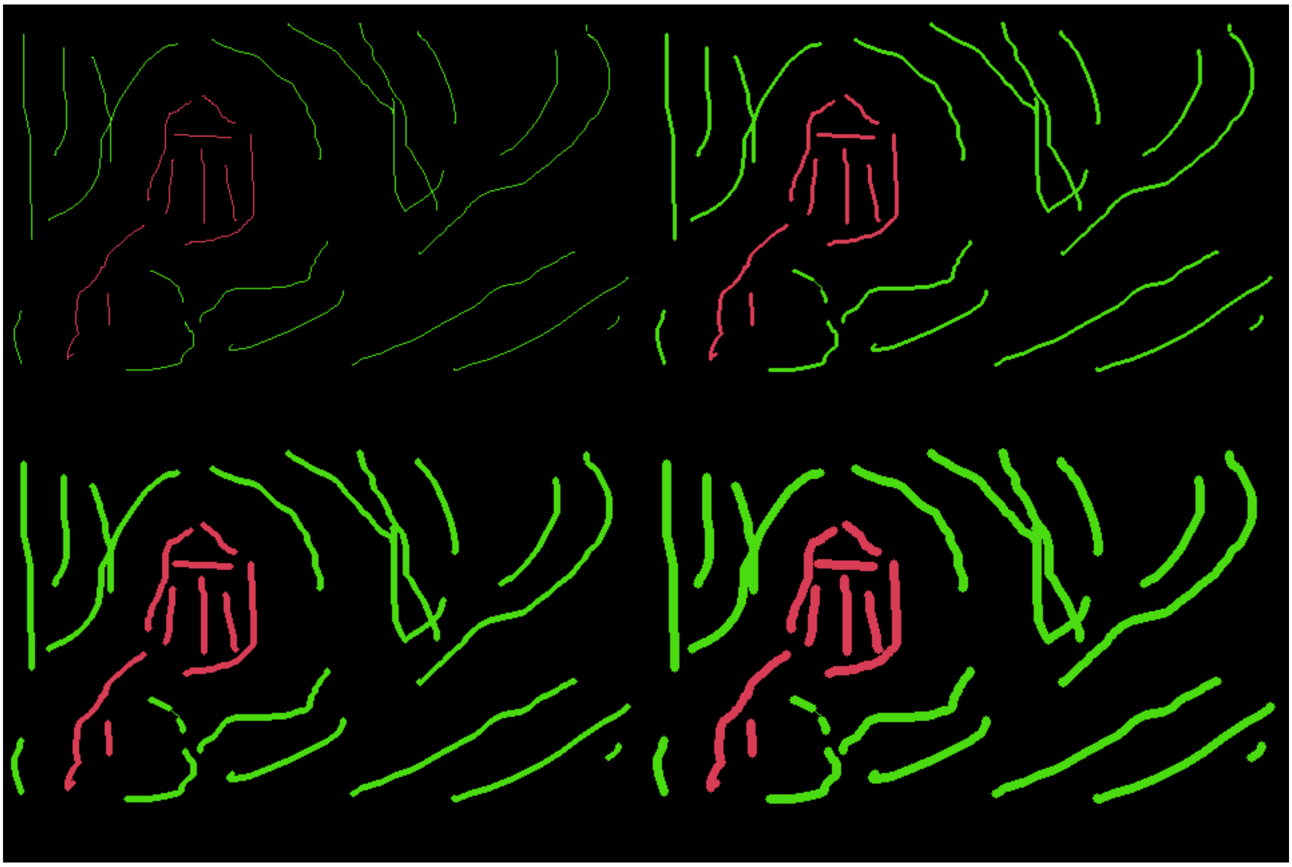

Figure 4: Dilation of radii 1-4 pixels on the label image in figure 3

example, if a pixel should be background but the user has marked it with a foreground stroke, then that pixel wouldn't be dilated. However, dilation still occurs for pixels that are correctly placed, but the dilation would spill over into the opposite label. Figure 3 shows an example of a user error here and figure 4 shows the resulting dilation. One can observe the gap that occurs where the background stroke overlaps with the foreground. The gap reduces with higher dilation radii due to dilation still occuring at the closest correctly labeled pixels (and dilating further into the error zone).

Ultimately, this makes little difference as errors tend to be very small. There are, however, a large number of these errors and typically we wouldn't use such an approach to dilate (as it requires knowing the right answer).

2.3 Segmentation programs

For the purpose of this study, we used Mark Eramian's implementation of the Boykov graph cut algorithm [4] as the primary algorithm of interest. This is largely because it is difficult to find ready-to-use algorithms for the purpose of this study and this one is tried and tested in several other studies that have taken place at the University of Saskatchewan [1][2].

We also used the OneCut algorithm [5] as an alternative algorithm to check our results against. We used a modified version [6] that allowed us to pass our label image in directly in the same manner as the Boykov graph cut algorithm. All images were separately run through both of the segmentation programs. The OneCut algorithm is somewhat of a side consideration as it seems to be extremely sensitive to the number of seeds and thus does not well handle many cases where users simply did not provide very many seeds (noticeable in how it handles points considerably worse than strokes). Figure 5 shows an example of this.

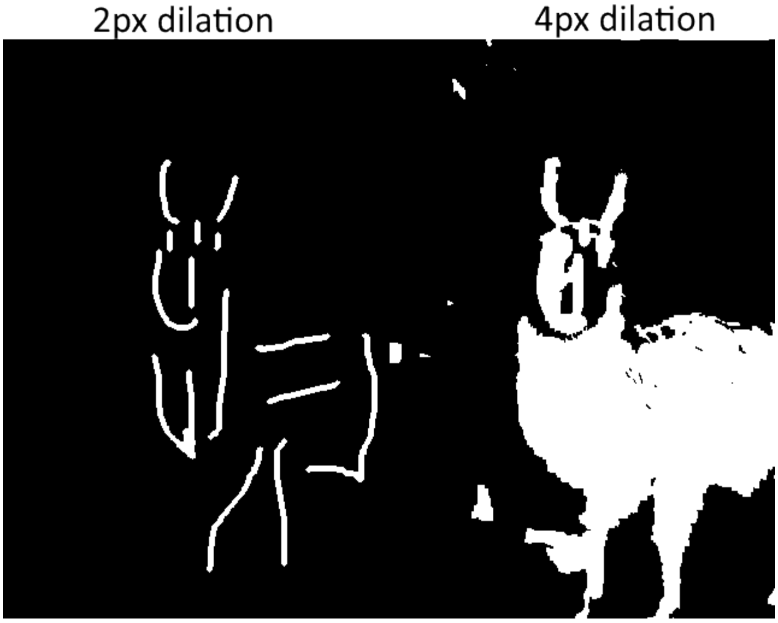

Figure 5: An example of OneCut's poor segmentation when it lacks enough seeds. The image is of a lama and the left case clearly did not segment anything of use – it merely returned the seeds.

3 Results

3.1 Methods of analysis

There's two main measures that we looked at in analyzing the segmented results. The accuracy and reproducibility.

We measure accuracy with the Dice Similarity Coefficient (DSC), which is defined as [7]:

$$\mathrm{DSC} = \frac{2|X \cap Y|}{|X| + |Y|}$$

Here, X and Y are the sets of pixels in the foregrounds of the image we're analyzing and its associated ground truth. Here, the notation $|X|$ denotes the sum of the binary matrix's elements (ie, how many pixels are in the foreground). The DSC in essence looks at the degree in common that the segmentation has with the ground truth compared to the total number of foreground pixels (thus, it's also considering when the segmentation has too many foreground pixels). Its value will be in between 0 and 1, where 0 implies no match at all and 1 implies a perfect match. Hence, higher values are better.

We measure reproducibility with the Generalized Tanimoto Coefficient (GTC), which is defined as [8]:

$$\mathrm{GTC} = \frac{\Sigma_i \left(X_i \wedge Y_i\right)}{\Sigma_i \left(X_i \vee Y_i\right)}$$

Here, X and Y are *all* matrices of binary segmentation of the same image across users (and thus X_i and Y_i are individual matrices, where $X_i \neq Y_i$). Or in other words, we're comparing all pairs of segmented images. What the GTC is essentially saying is that we're looking at what all users got in common (for foreground) and dividing it by what all users had. That tells us how similar the users results were. The ideal segmentation program, after all, would give us similar results (ideally perfect ones) for all reasonable user annotations. We don't want the segmentation program to be vulnerable to extreme inconsistency in results due to exactly where the user places their annotations. The GTC is also a value between 0 and 1, where 0 implies that users all got completely different results and 1 implies they all got the exact same results. Again, higher values are better.

3.2 Effects on accuracy

Both the mean and median DSC rose in all of our groups of independent variables when comparing undilated results to 4 pixels of dilation. Figures 6 through 9 show these results. We can observe that mean DSC increases between 0.010 to 0.065 when comparing no dilation to 4 pixels of dilation. There's occasional drops in smaller dilation radii for unknown reasons. It seems to imply some wild inconsistency in how well certain images get segmented (and certainly some are consistently segmented well and others are frequently poorly segmented). Some users simply don't provide as many annotations as others, which is a major cause for some images to segment poorly for them, yet segment well for other users. This only happens for the smaller time pressures in Yuanxia's study, implying that perhaps the placement of annotations by hurried users is simply very poor.

There's some atypical results shown in 9, with the median values actually dipping for moderate dilation values, but still ultimately showing an increase between undilated and 4 pixels of dilation. It can be seen from examining the results that sometimes the 1-3 pixel dilation actually has introduced new errors, but it's unclear why this happens (a quirk of the algorithm?).

These results are not normally distributed, so the Friedman test was applied within each independent variable (that is, it was applied to all of the stroke images in Rau's study separately from the point images, etc). This found that all the differences are statistically significant, $p < 0.01$.

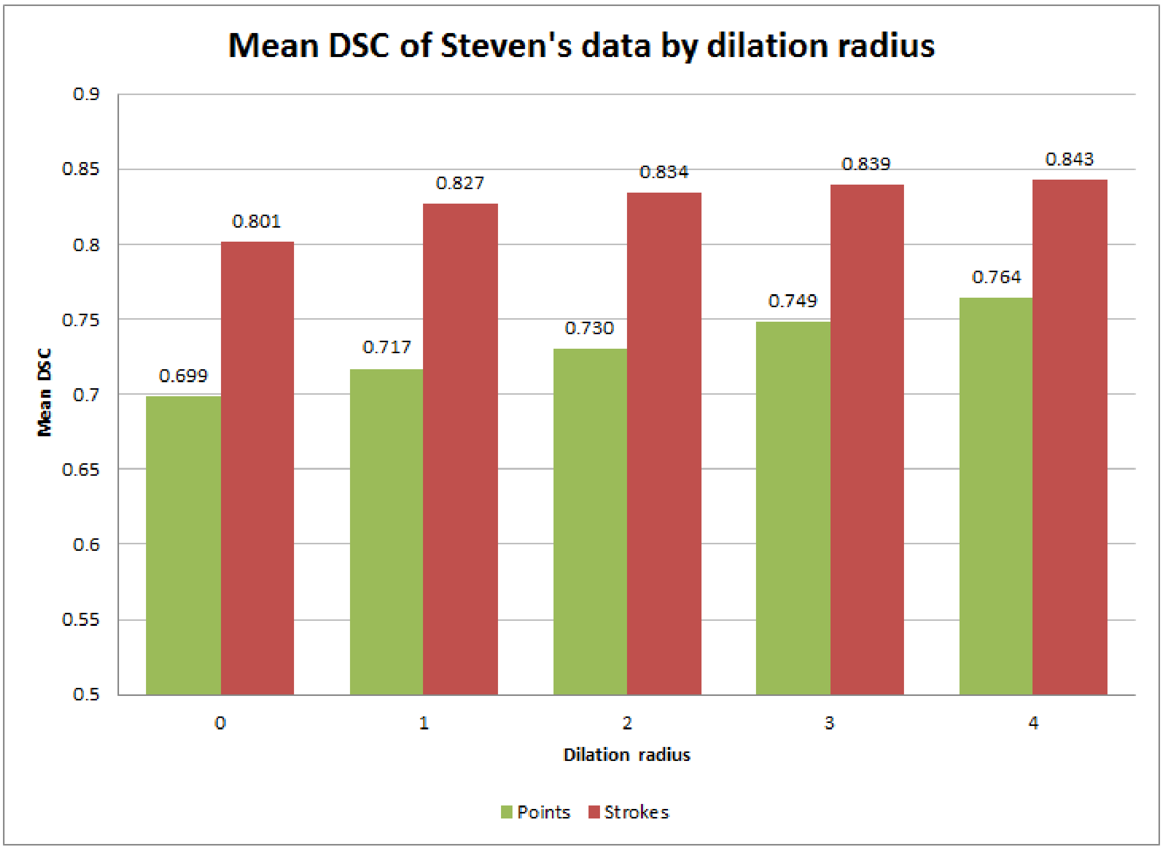

Figure 6: Mean accuracy in Steven Rau's study.

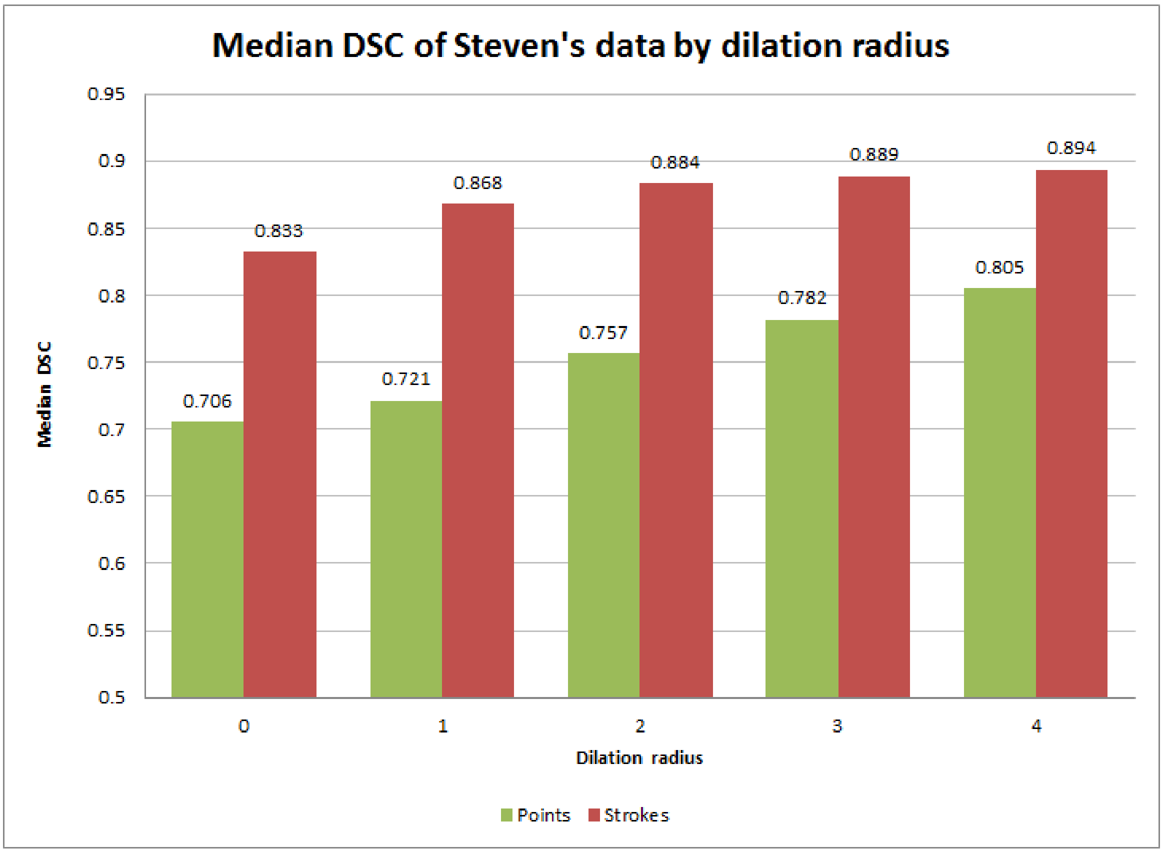

Figure 7: Median accuracy in Steven Rau's study.

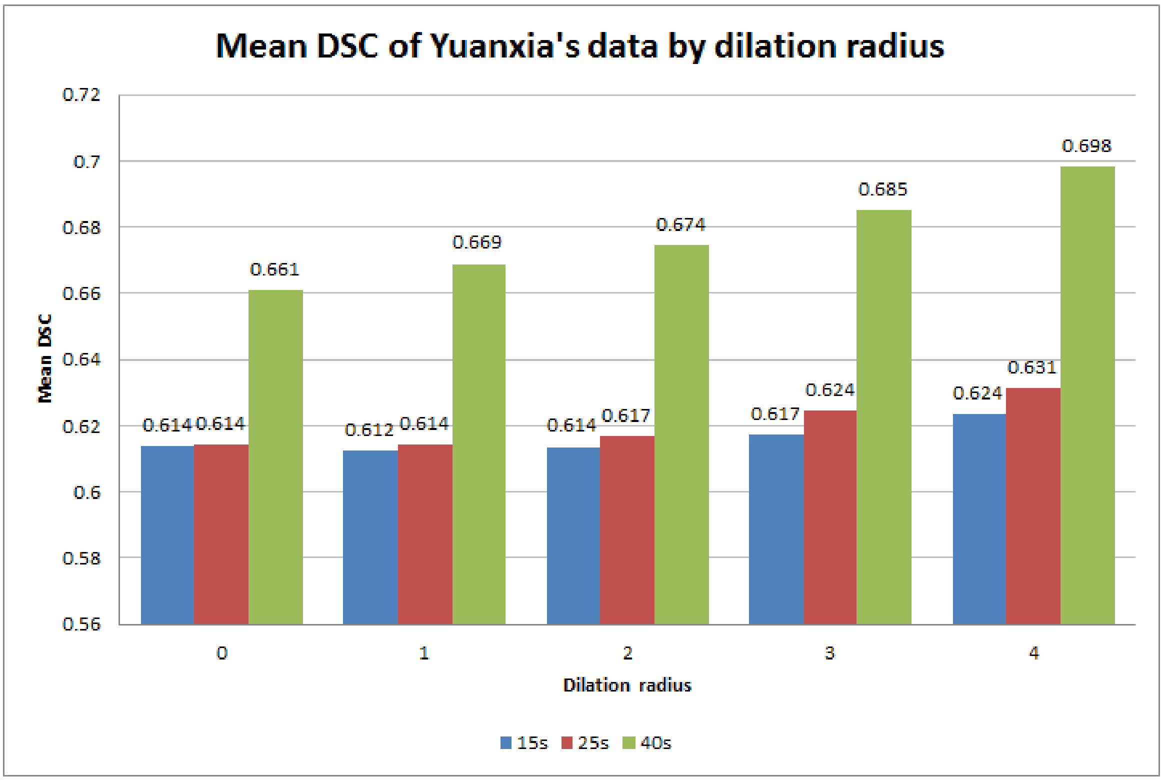

Figure 8: Mean accuracy in Yuanxia Li's study.

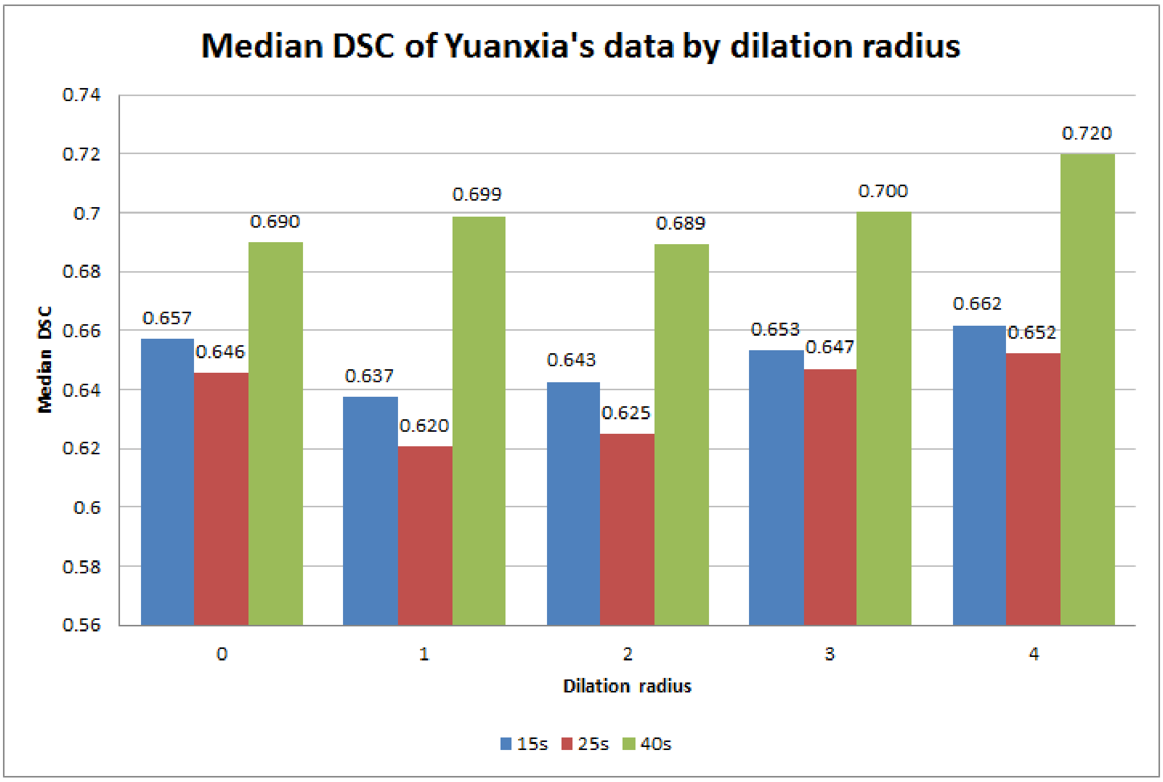

Figure 9: Median accuracy in Yuanxia Li's study.

3.3 Effects on reproducibility

The mean and median GTC has also risen when comparing undilated results to the 4 pixel dilation results. The mean GTC has increased between 0.061 and 0.069 when comparing no dilation versus 4 pixels of dilation. This shows that the increased dilation has made our results more likely to be similar. Figures 10 through 13 show these results.

There's some notable inconsistency in figure 13. We note that the GTC of the 15 second time pressure group has actually peaked at smaller dilation radii. The most likely explanation here is simply that rushed-for-time users were very bad at placing annotations and thus obtained some rather wildly varying results.

The Friedman test was also applied to this statistic and found that all the differences are statistically significant, $p < 0.01$.

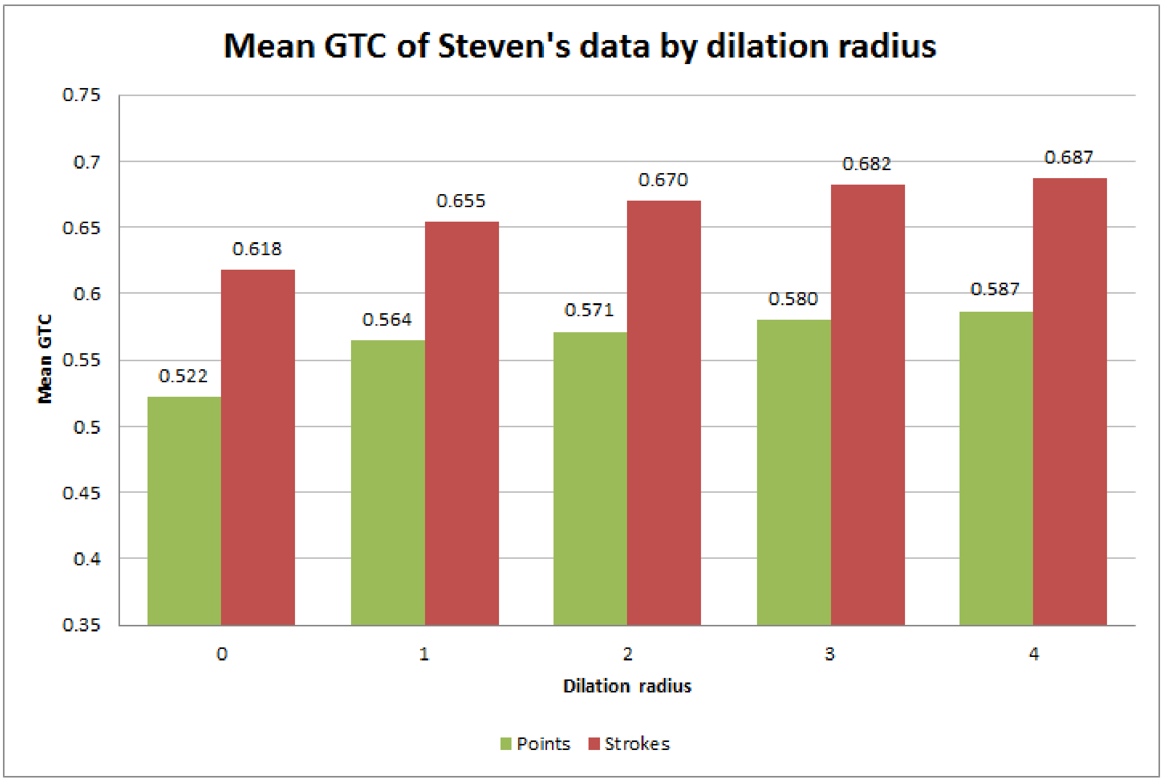

Figure 10: Mean reproducibility in Steven Rau's study.

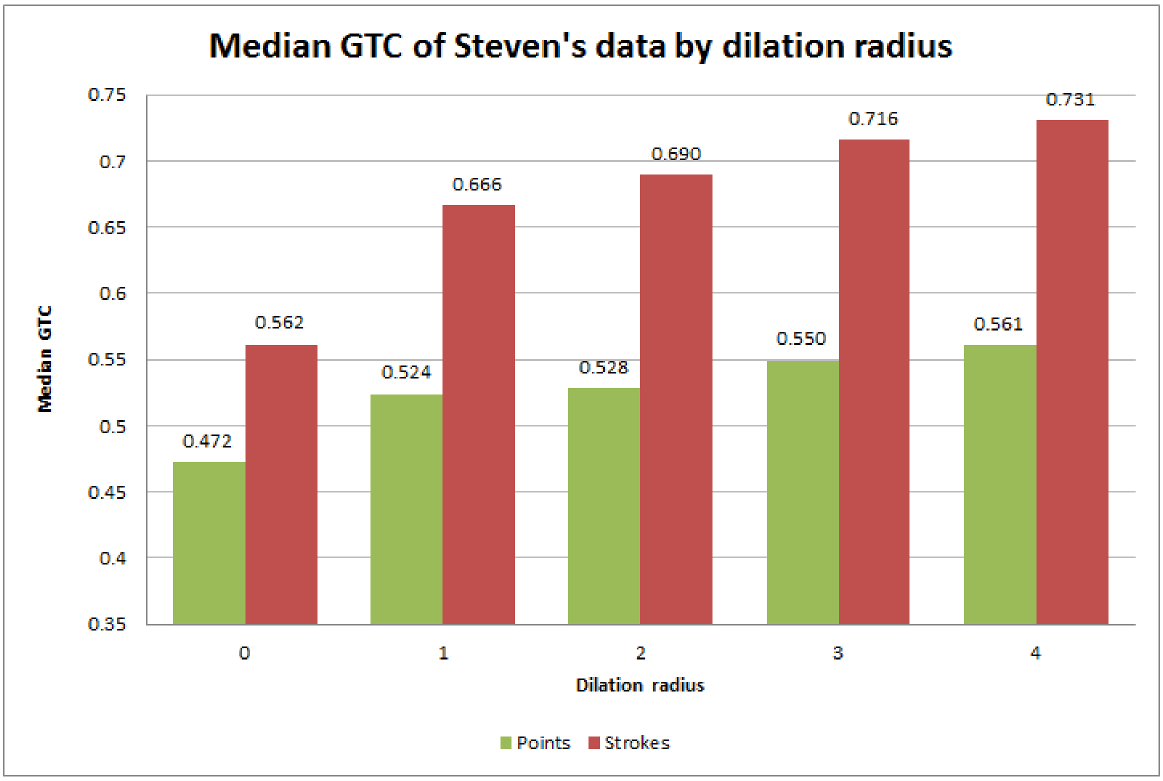

Figure 11: Median reproducibility in Steven Rau's study.

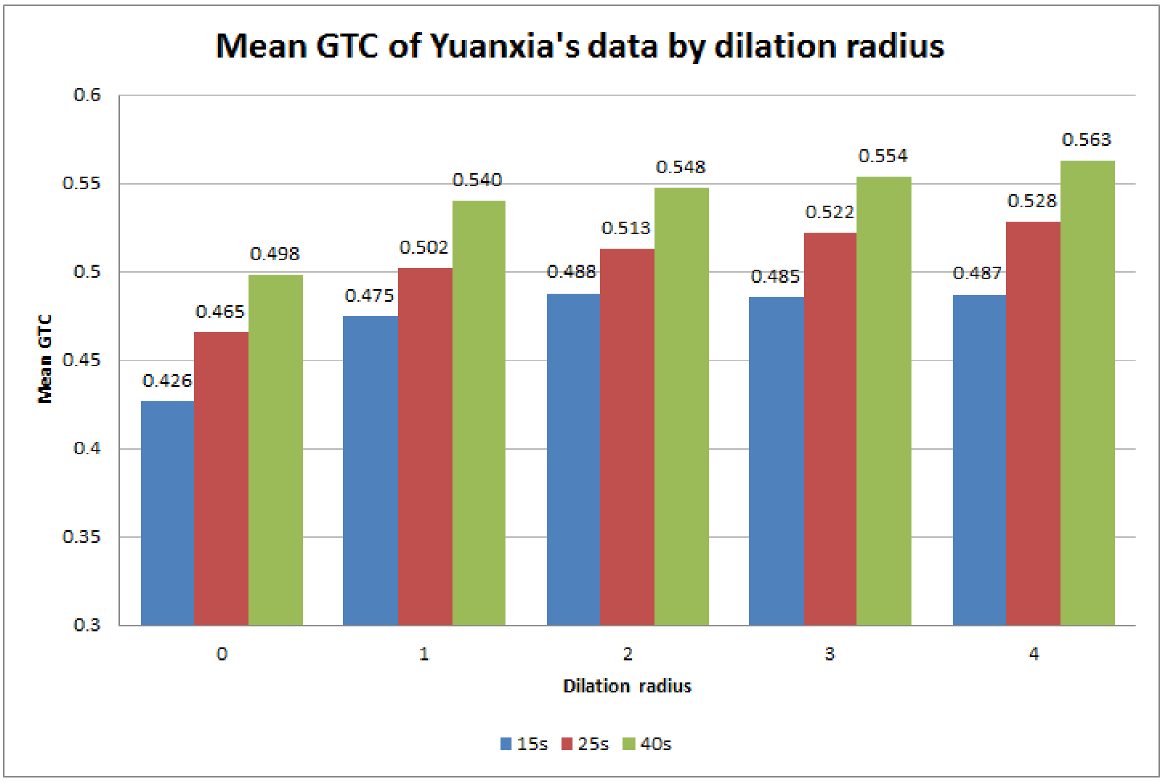

Figure 12: Mean reproducibility in Yuanxia Li's study.

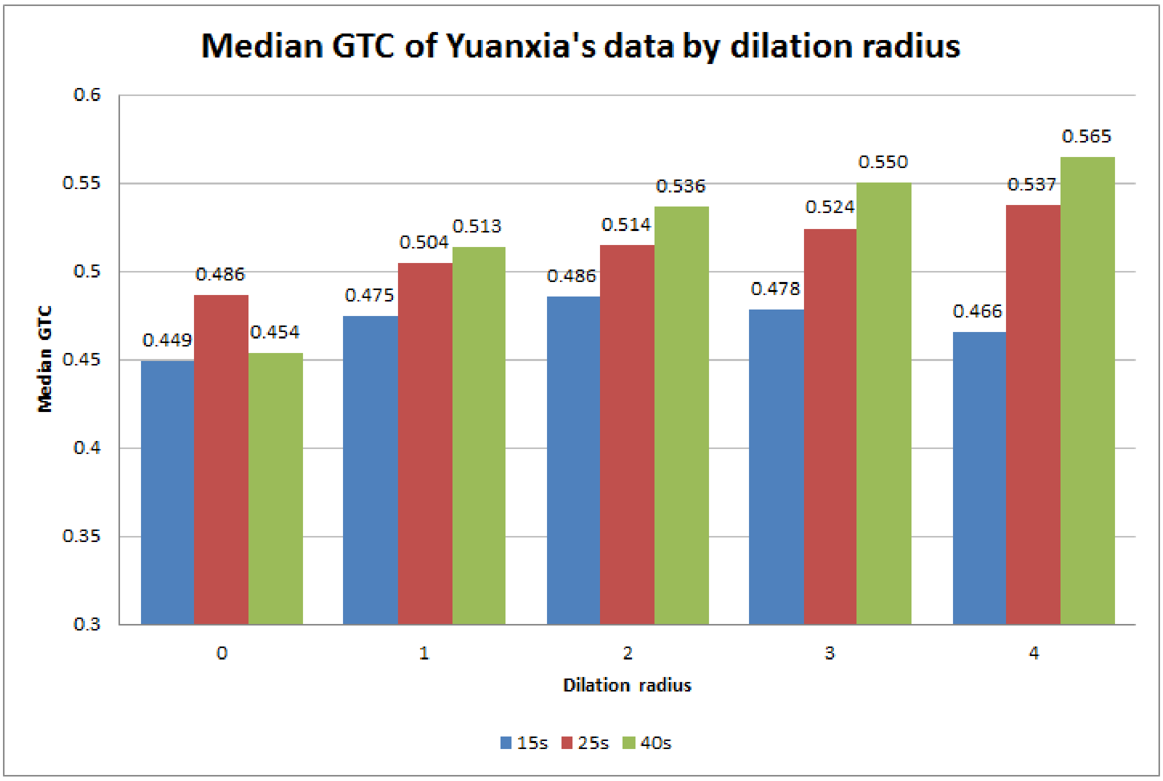

Figure 13: Median reproducibility in Yuanxia Li's study.

3.4 Effects with other segmentation algorithms

We can see in figures 14 and 15 how poorly the OneCut algorithms perform in terms of accuracy. There are some images that they do well at, but many they simply fail to segment in any meaningful way. We can only look so far at this, however, because users were trained on the Boykov graph cut algorithm and thus were working on the expectation that they needed to provide annotations that worked for it. The OneCut algorithm simply seems to need a considerably larger number of seed points to do well. This is evidenced from the massive increases in DSC as the dilation increases. We have to remember, however, that many images aren't actually being segmented (see figure 5) and thus the dilation naturally will increase the DSC of those images, but without providing any telling results.

The Friedman test showed that the differences in DSC and GTC between the various amounts of dilation (0-4 pixels) were significant with the OneCut algorithm, between all independent variables, $p < 0.01$.

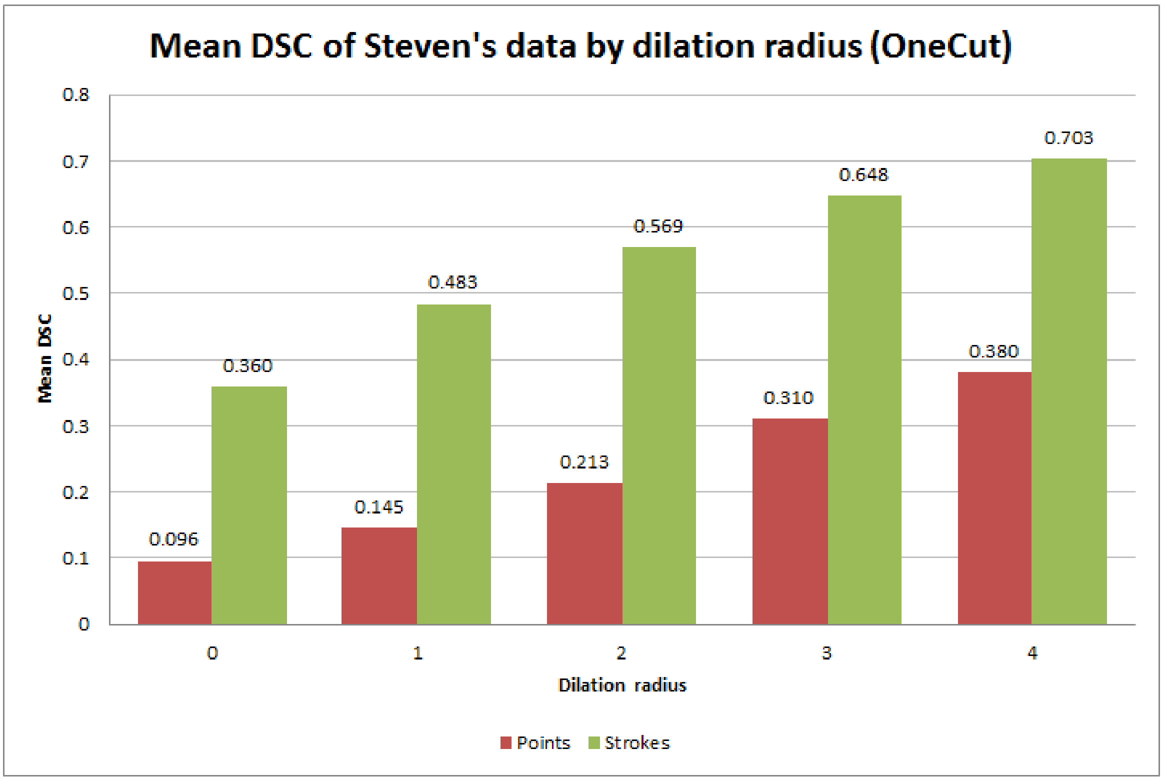

Figure 14: Mean OneCut accuracy in Steven Rau's study.

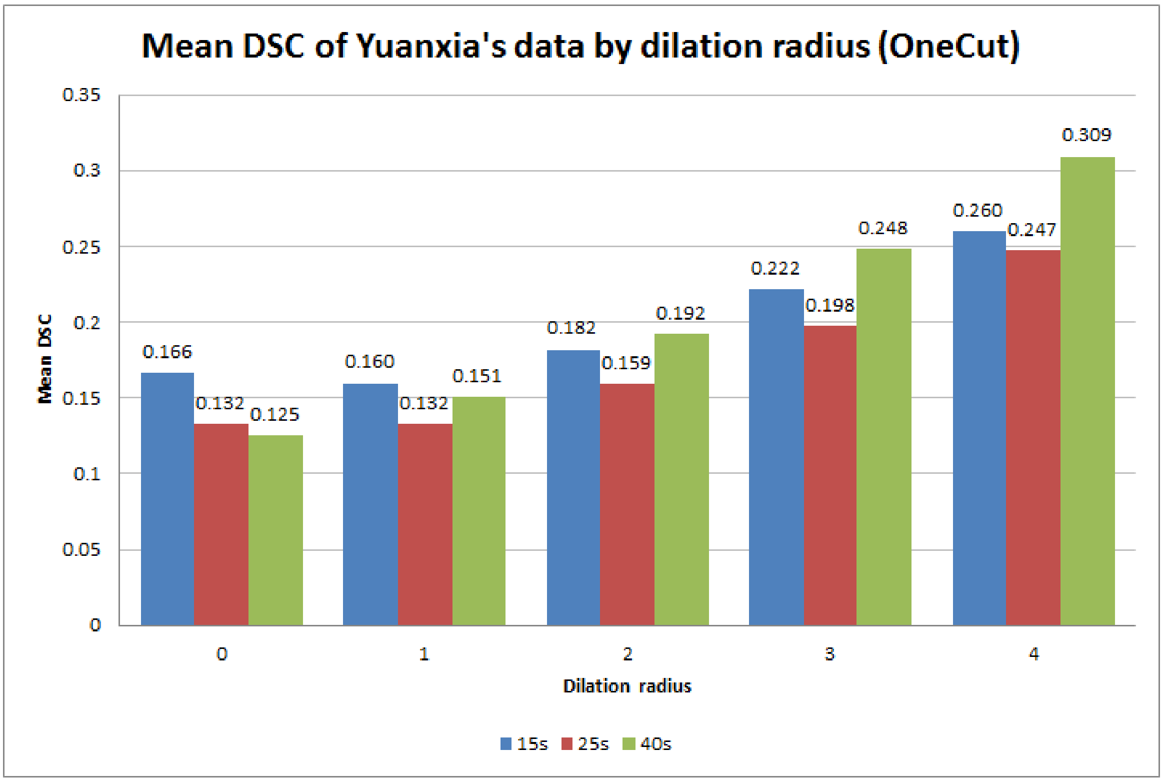

Figure 15: Mean OneCut accuracy in Yuanxia Li's study.

The OneCut algorithm's reproducibility seems normal with Rau's data, as we see in figure 16 (if shockingly low – showing how poorly the algorithm performed with many images). But with Li's data, in figure 17, we can see that the results were extremely inconsistent from the past trend. Lower time pressures were actually more consistent (higher GTC) and dilation didn't help them so much. It seems that this actually happens because the algorithm occasionally has the very odd behavior of sometimes "segmenting" in such a way that everything except the pixels marked as background is considered foreground (and hence resulting in images that are 90% foreground). This seems to happen less frequently with strokes than points. As well, the higher time pressures have it happening less often (presumably because they have more points). Naturally having almost all the image being considered foreground would be considered very consistent (if flat out wrong). This is also a good example of why we don't want to look at the GTC in a vacuum, as we can have results that are consistently wrong scoring a high GTC.

The Friedman test showed that all of these results *except* for that of Li's 40 second group were significant, $p < 0.01$.

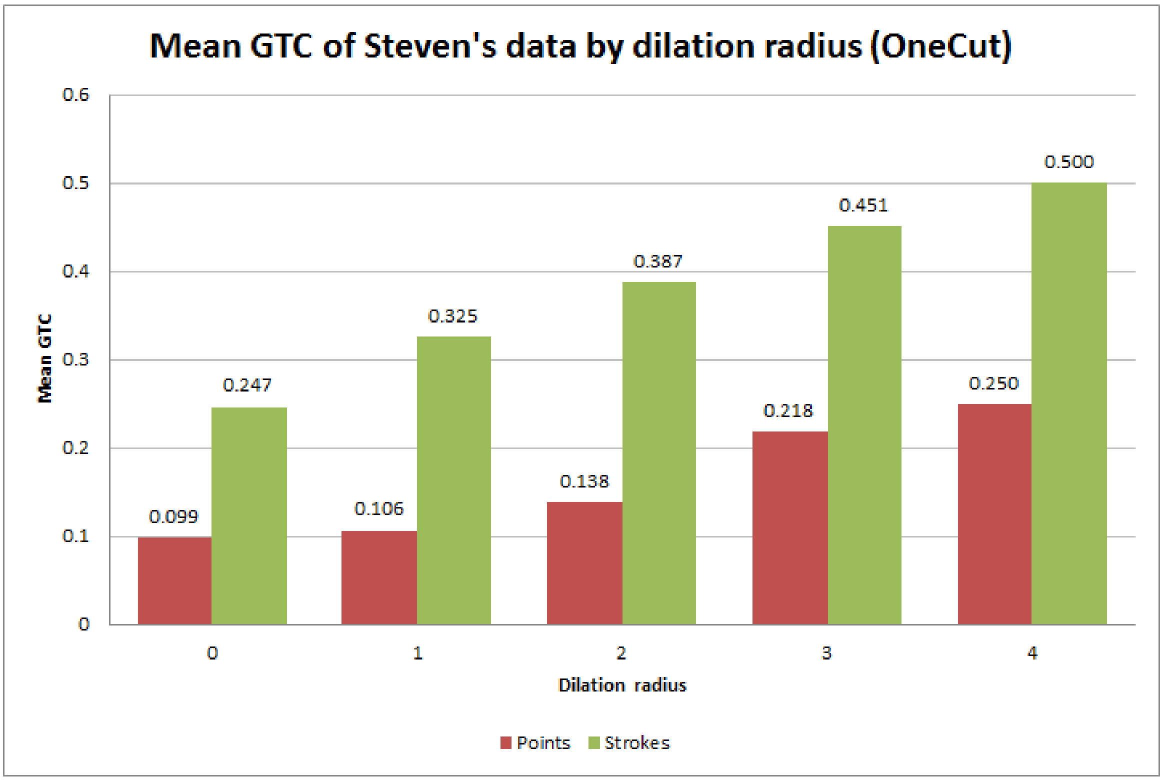

Figure 16: Mean OneCut reproducibility in Steven Rau's study.

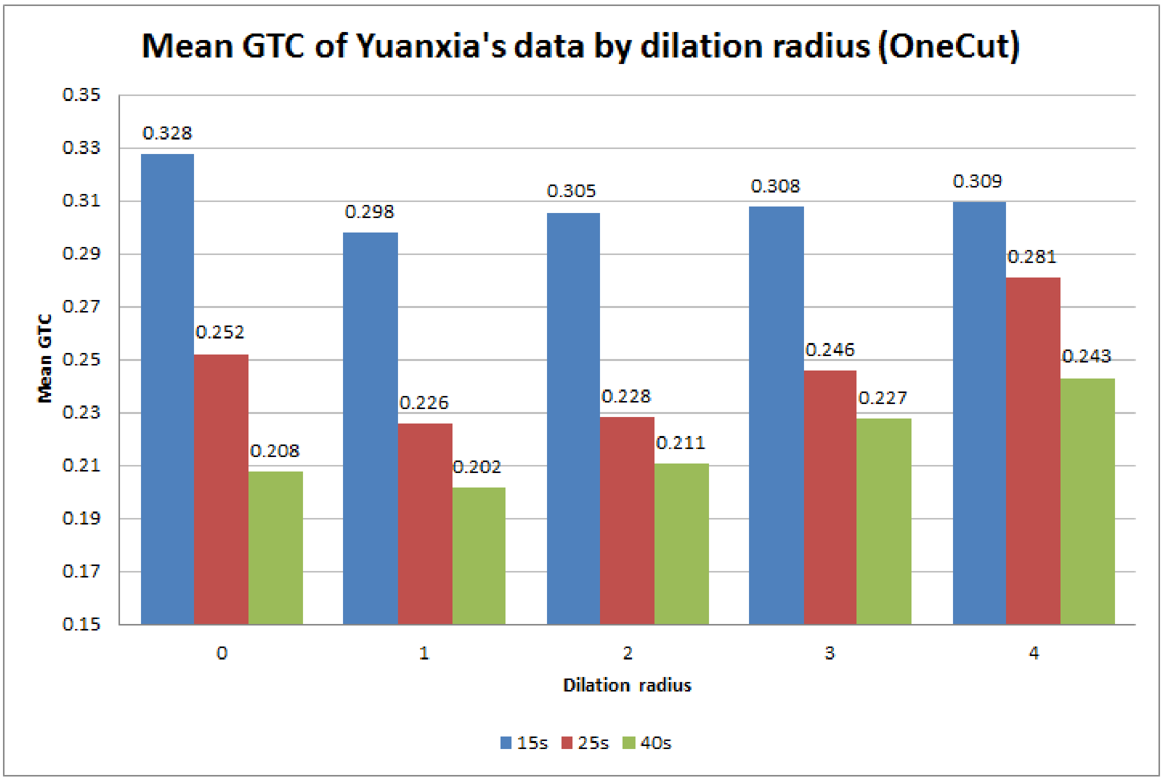

Figure 17: Mean OneCut reproducibility in Yuanxia Li's study.

4 Conclusion and discussion

The results are not always completely consistent, but in general, we can conclude that the difference between no dilation and 4 pixels of dilation is a statistically significant increase in accuracy and reproducibility (certainly at least with the Boykov algorithm). In general, we get better results with dilation if we allow users time to place annotations, as we observe that low time pressures sometimes get more finnicky results. It was known from Rau's study that strokes perform better than points [1], which was surely due to the larger number of seeds. But we can also see that we can go further, still.

The results can be highly variable by user, however. Some users simply do not provide enough annotations for segmentation to work well. Dilation can improve this, but can only do so much. Users simply spending more time and effort to provide a larger set of annotations makes a considerable difference both with and without the dilation. Arguably software that utilizes semi-automatic image segmentation in this technique should not only use thicker strokes, but also ensure that users are aware of the impact of simply adding more annotations.

One notable question is if we might get different results if users were aware of the larger "pen size". Might they make fewer errors and redundancies? This would also avoid the need for our dilation algorithm to utilize a non-traditional approach in trying to avoid introducing new errors. There's the worry that this approach could create unrealistic results.

The OneCut algorithm that we tested proved interesting. In general, it should probably be regarded as little more than a side consideration in this study. There's the big issue that users were not trained with this algorithm and thus did not realize that it is much more sensitive to how much information is provided. As well, it just plain works very poorly on undilated label images. It was clearly not created with so few seeds in mind and the Rau and Li studies were not designed with that requirement. Hence, the provided user data is insufficient to well test it, especially when we have so many cases of no segmentation occuring, which essentially ends up statistically drowning out relevant results. However, we can at least conclude that the OneCut algorithm is extremely sensitive to the amount of annotations and thus recommend that its users provide many, thick strokes if they wish to get meaningful results.

The fact that semi-automatic image segmentation algorithms seem sensitive the quantity of seed points (which is really what dilating the strokes is doing) also serves to partially explain the results of strokes versus points. Stroke annotations naturally provide more seed points than point annotations, as dilation does. That said, strokes are much more efficient than points because they provide more *useful* annotations (due to greater range of area). And as we've shown, thicker strokes perform better still.

The code and data for this study are available online at `https://github.com/KatrinaHoffert/stroke-radius-segmentation` [9].

References

[1] S. Rau and M. Eramian, “Analysis of user input methods for semi-automatic segmentation,” Undergrad thesis, University of Saskatchewan.

[2] Y. Li and M. Eramian, “Effects of time pressure on semiautomatic segmentation,” University of Saskatchewan, Tech. Rep., unpublished.

[3] Berkeley Computer Vision Group. Berkeley segmentation data set and benchmarks 500 (BSDS500). Accessed 2017-03-28. [Online]. Available: https://www2.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/resources.html#bsds500

[4] Y. Boykov and G. Funka-Lea, “Graph cuts and efficient n-d image segmentation,” *International Journal of Computer Vision*, vol. 70, no. 2, pp. 109–131, 2006. [Online]. Available: http://link.springer.com/article/10.1007/s11263-006-7934-5

[5] M. Tang, L. Gorelick, O. Veksler, and Y. Boykov, “Grabcut in one cut,” *in IEEE Conference on Computer Vision (ICCV)*, 2013. [Online]. Available: http://ieeexplore.ieee.org/document/6751330/

[6] K. Hoffert. Onecut-cli on github. Accessed 2017-03-28. [Online]. Available: https://github.com/KatrinaHoffert/OneCut-cli

[7] L. R. Dice, “Measures of the amount of ecologic association between species,” *Ecology*, vol. 26, no. 3, pp. 297–302, 1945. [Online]. Available: http://www.jstor.org/stable/1932409

[8] T. Tanimoto, *An Elementary Mathematical Theory of Classification and Prediction*. International Business Machines Corporation, 1958.

[9] K. Hoffert. Study code and data. Accessed 2017-03-28. [Online]. Available: https://github.com/KatrinaHoffert/stroke-radius-segmentation