DD2423
Lab III
Alexandros Filotheou

1 K-means clustering

1.1 Question 1

The result of the initialization of the clustering process depends on whether there is a priori knowledge about the image to be segmented. In this example, where image `orange` is known, we can specify the colour of each initial cluster depending on what the image's dominant colours are. In this case where the colour diversity is low, it is possible to set two initial kernel values to the colours "orange" (roughly $(255, 150, 0)$ in the image in question) and "white" (exactly $(255, 255, 255)$ in the image in question) and let the rest of the kernels be chosen in random, to the extent of what level of detail is desirable.

In general, where the image to be segmented and its context may be unknown, the best way to initialize the kernels, so as to achieve a result that is not wrongly biased, is to let them be chosen at random, optinally with provision having been taken regarding the diversity of initial colours.

Figures 1, 2 and 3 illustrate the k-means segmentation method for image `orange` with

$$\texttt{(K,L,scale\_factor, } \sigma\texttt{)} \equiv \texttt{(8,10, 1, 1)}$$

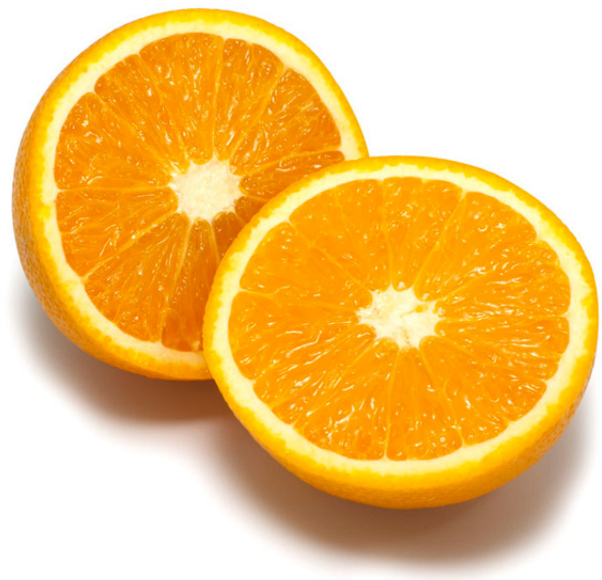

Figure 1: Image `orange`.

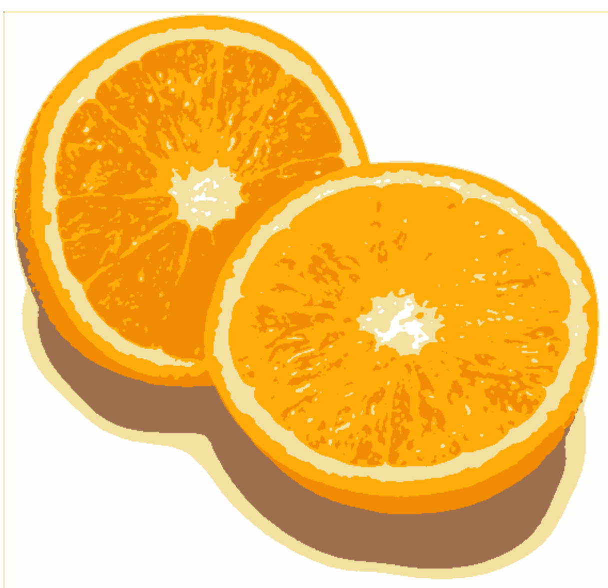

Figure 2: Image `orange` segmented.

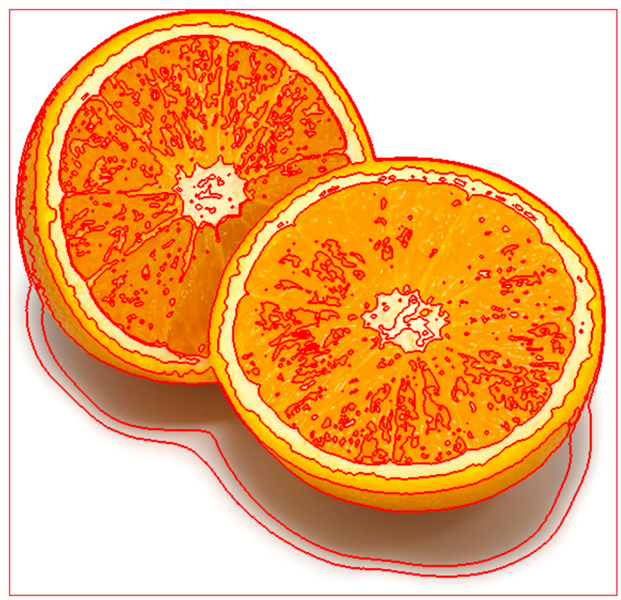

Figure 3: Image `orange` and the bounds of its segments.

1.2 Question 2

Convergence time depends linearly on the number of clusters K chosen and the size of the image. Furthermore it depends on the colour diversity of the image and is inversely proportional to the amount of pre-blurring. Tables 1 and 2 illustrate the number of iterations it takes for the `K-means` clustering method to reach convergence for images `orange` and `tiger1` for different number of clusters.

K	convergence iteration
1	2
2	9
3	14
4	11
5	18
6	21
7	17
8	35
9	46
10	23
11	38
12	31

Table 1: Number of centers K and number of iterations taken to reach convergence with regard to image `orange`. `(scale_factor,` σ`)` $\equiv$ `(1, 1)`

K	convergence iteration
1	2
2	12
3	31
4	83
5	67
6	52
7	57
8	208
9	90
10	103
11	124
12	132

Table 2: Number of centers K and number of iterations taken to reach convergence with regard to image `tiger1`. `(scale_factor,` σ`)` $\equiv$ `(1, 1)`

1.3 Question 3

If I understand the question correctly, this value is $K = 13$ for image `orange`. Figures 4 and 5 illustrate that a superpixel covers parts from both halves of the orange for $K = 12$, whereas in figures 6 and 7, where $K = 13$, there is a clear boundary between them.

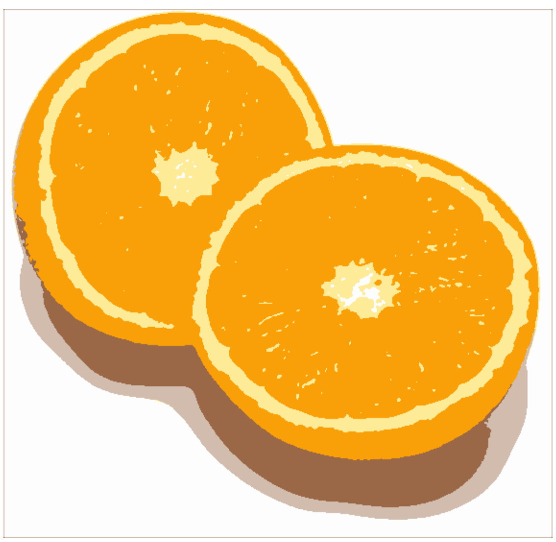

Figure 4: Image `orange` segmented using $K = 12$ clusters.

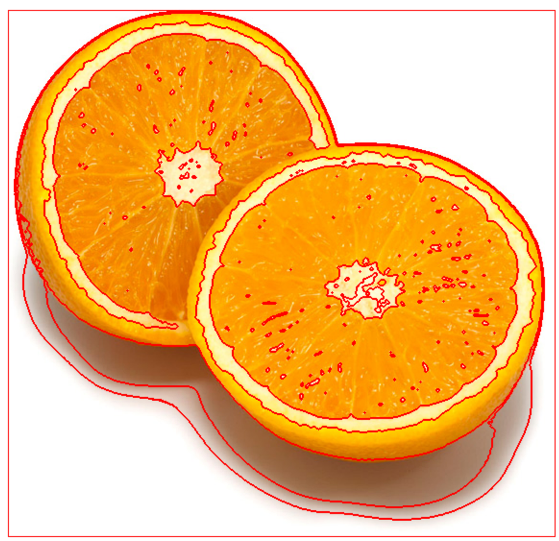

Figure 5: Image `orange` and the bounds of its segments for $K = 12$.

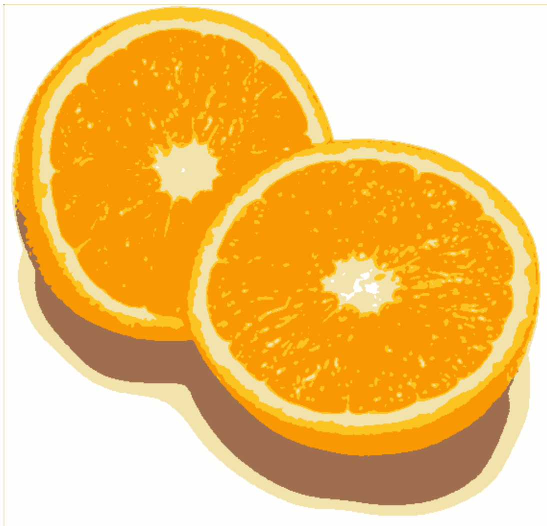

Figure 6: Image `orange` segmented using $K = 13$ clusters.

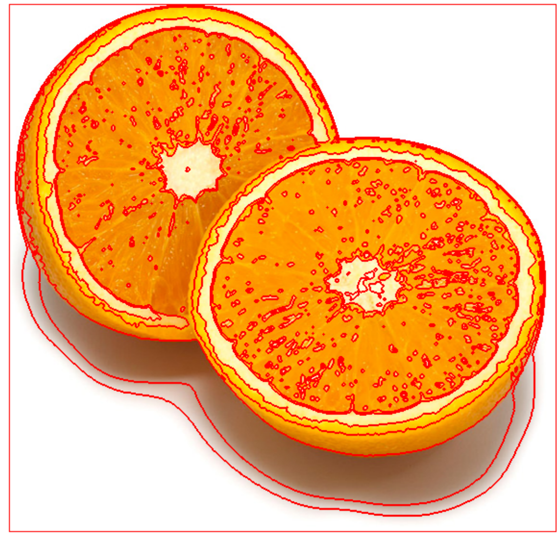

Figure 7: Image `orange` and the bounds of its segments for $K = 13$.

1.4 Question 4

Initially, what should be done is to increase the number of clusters K since images `tiger{1,2,3}` is more diverse in colour than image `orange`. As convergence takes more iterations to be met for increasing number of clusters, the iteration upper threshold L should be increased as well.

In the case where the clusters are not initialized in random but with a certain colour set that is desired to be achieved, then the centers of these clusters should be set to the values of those clusters.

2 Mean-shift segmentation

2.1 Preliminary results

2.1.1 Image **orange**. Varying σ_s^2.

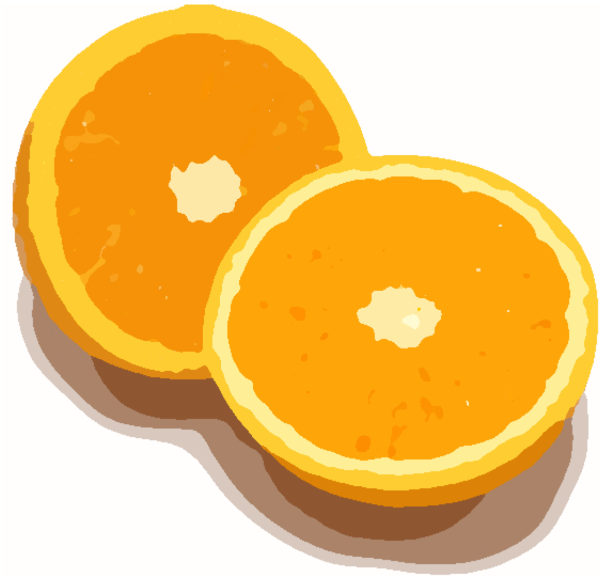

Figure 8: Image **orange** segmented with **mean-shift**. $(\sigma_s^2, \sigma_c^2) \equiv (3.0, 4.0)$

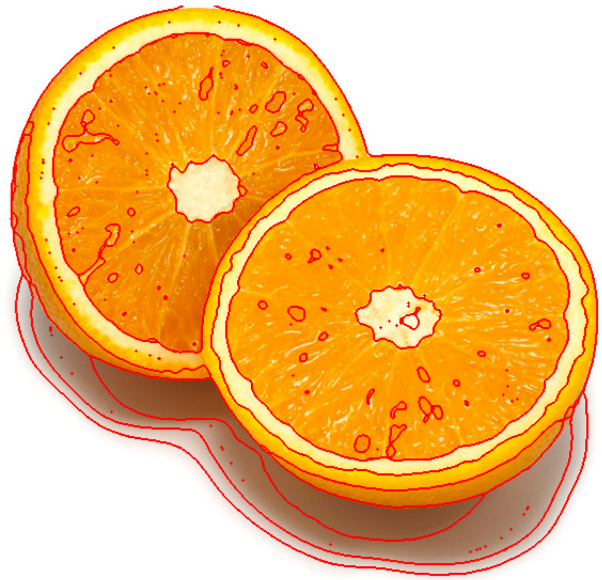

Figure 9: Segmentation bounds of image **orange**. $(\sigma_s^2, \sigma_c^2) \equiv (3.0, 4.0)$

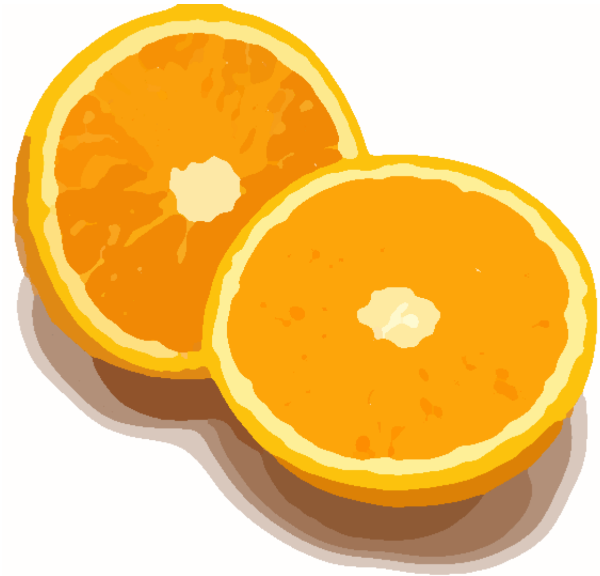

Figure 10: Image **orange** segmented with **mean-shift**. $(\sigma_s^2, \sigma_c^2) \equiv (4.0, 4.0)$

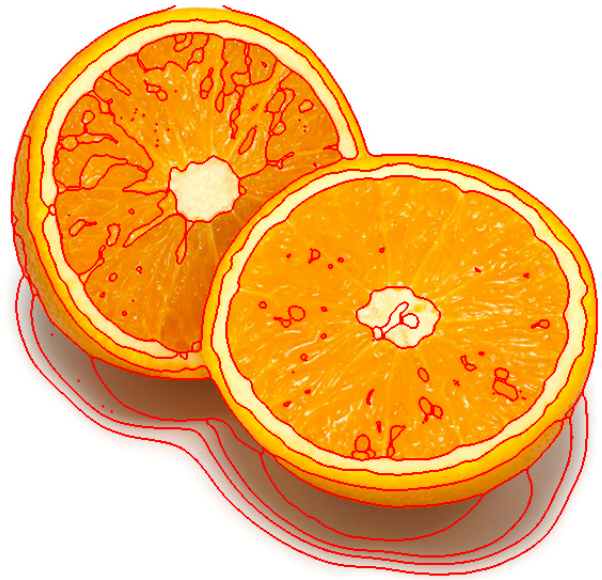

Figure 11: Segmentation bounds of image **orange**. $(\sigma_s^2, \sigma_c^2) \equiv (4.0, 4.0)$

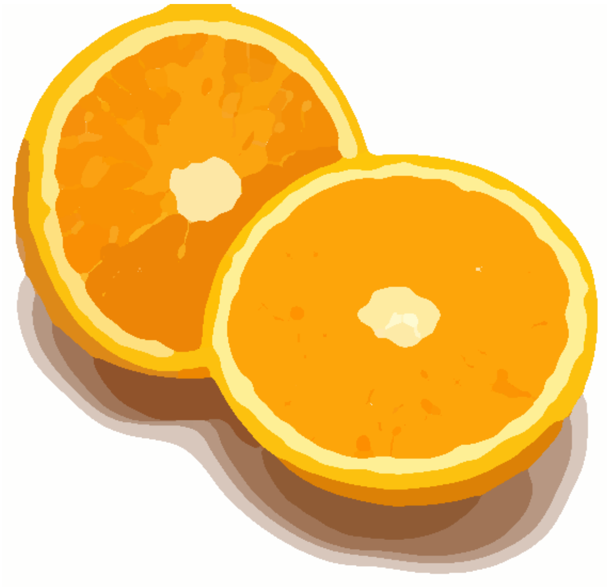

Figure 12: Image `orange` segmented with `mean-shift`. $(\sigma_s^2, \sigma_c^2) \equiv (5.0, 4.0)$

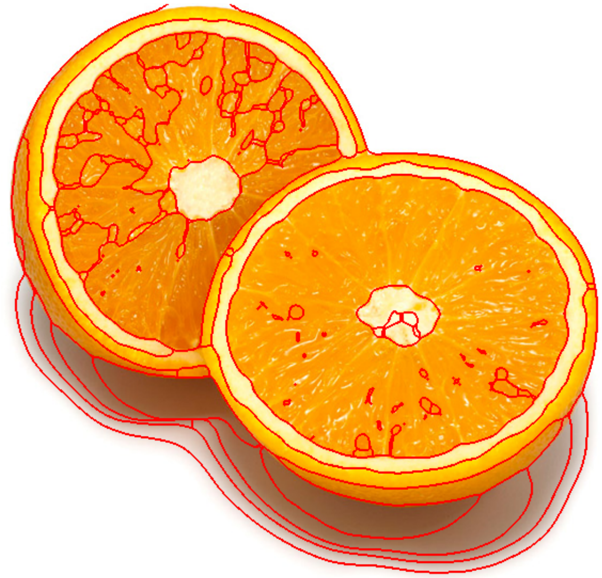

Figure 13: Segmentation bounds of image `orange`. $(\sigma_s^2, \sigma_c^2) \equiv (5.0, 4.0)$

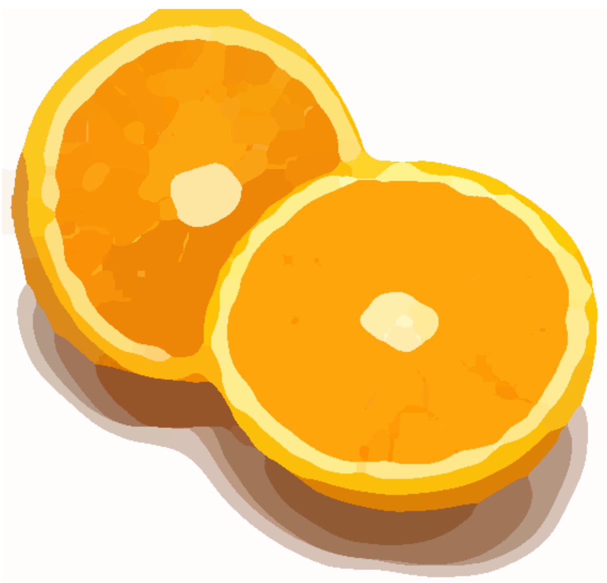

Figure 14: Image `orange` segmented with `mean-shift`. $(\sigma_s^2, \sigma_c^2) \equiv (8.0, 4.0)$

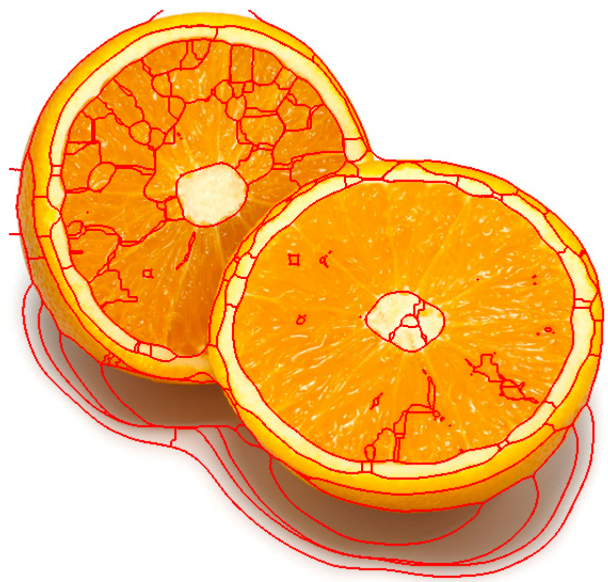

Figure 15: Segmentation bounds of image `orange`. $(\sigma_s^2, \sigma_c^2) \equiv (8.0, 4.0)$

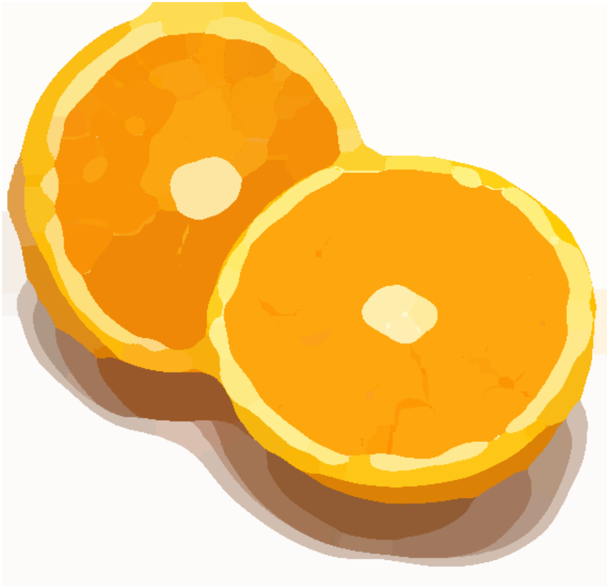

Figure 16: Image `orange` segmented with `mean-shift`. $(\sigma_s^2, \sigma_c^2) \equiv (10.0, 4.0)$

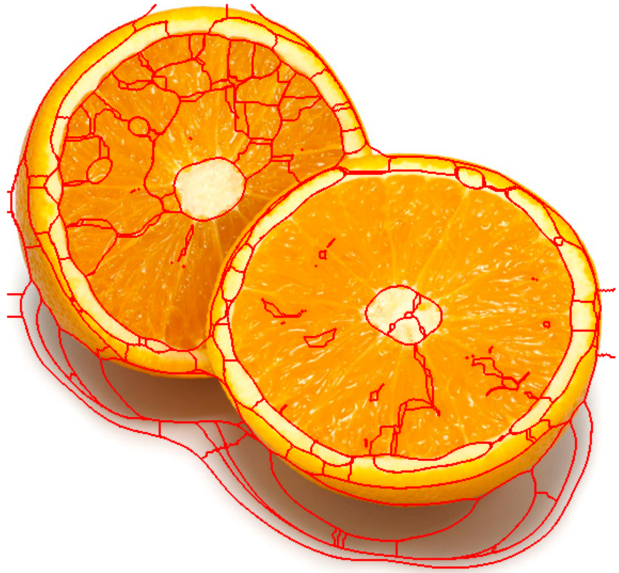

Figure 17: Segmentation bounds of image `orange`. $(\sigma_s^2, \sigma_c^2) \equiv (10.0, 4.0)$

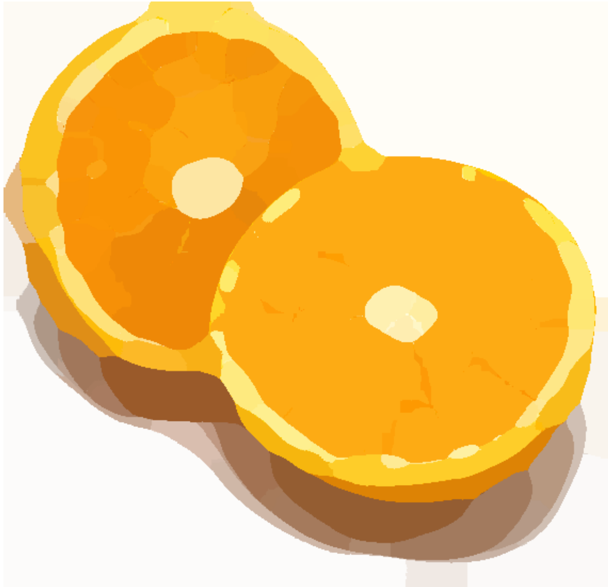

Figure 18: Image `orange` segmented with `mean-shift`. $(\sigma_s^2, \sigma_c^2) \equiv (12.0, 4.0)$

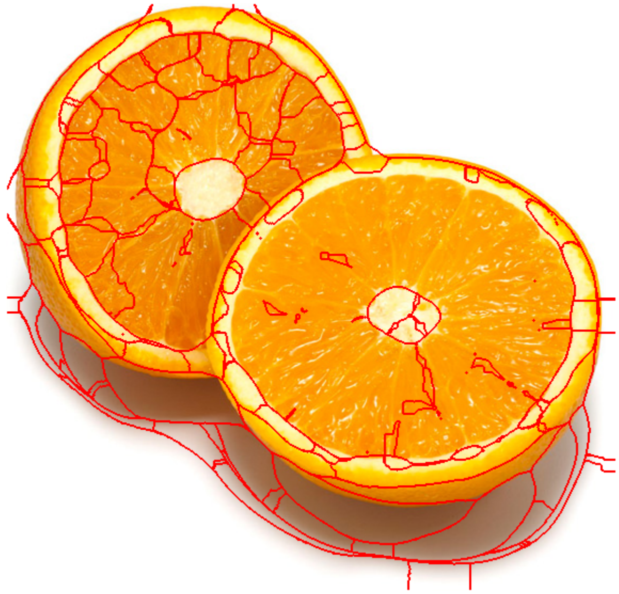

Figure 19: Segmentation bounds of image `orange`. $(\sigma_s^2, \sigma_c^2) \equiv (12.0, 4.0)$

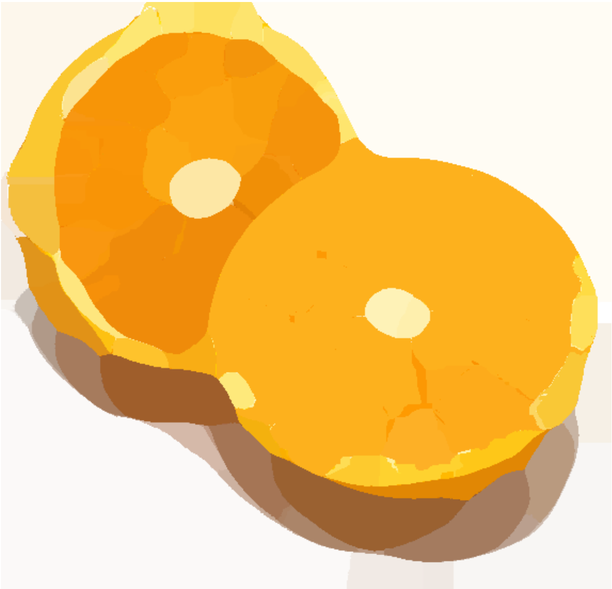

Figure 20: Image `orange` segmented with `mean-shift`. $(\sigma_s^2, \sigma_c^2) \equiv (15.0, 4.0)$

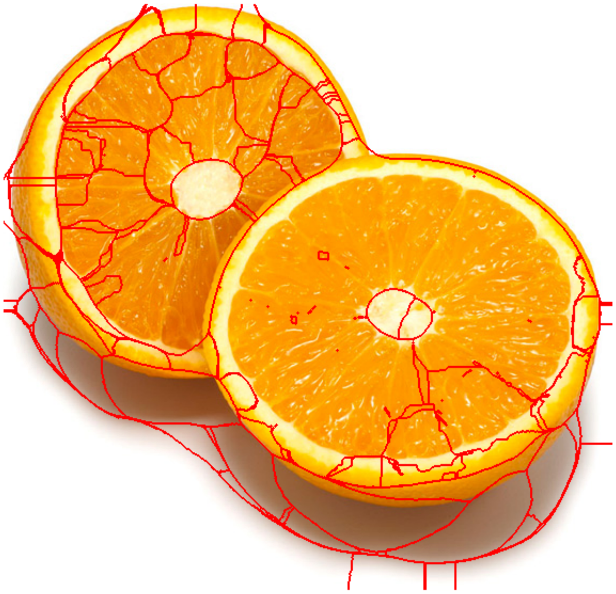

Figure 21: Segmentation bounds of image `orange`. $(\sigma_s^2, \sigma_c^2) \equiv (15.0, 4.0)$

Figure 22: Image `orange` segmented with `mean-shift`. $(\sigma_s^2, \sigma_c^2) \equiv (20.0, 4.0)$

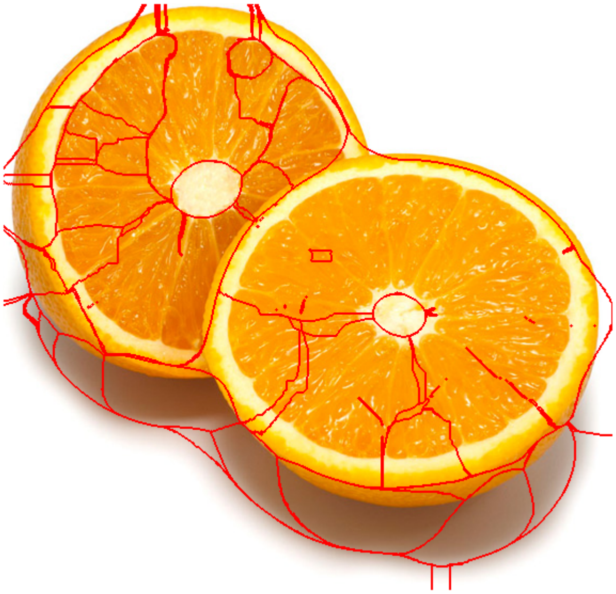

Figure 23: Segmentation bounds of image `orange`. $(\sigma_s^2, \sigma_c^2) \equiv (20.0, 4.0)$

2.1.2 Image **orange**. Varying σ_c^2.

Figure 24: Image `orange` segmented with `mean-shift`. $(\sigma_s^2, \sigma_c^2) \equiv (8.0, 1.0)$

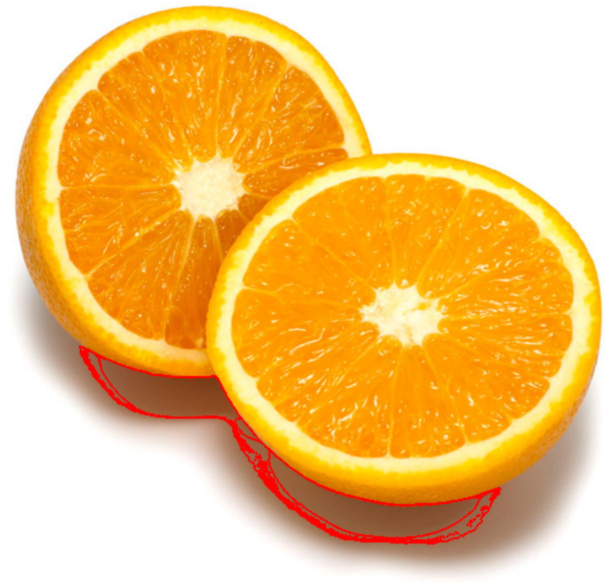

Figure 25: Segmentation bounds of image `orange`. $(\sigma_s^2, \sigma_c^2) \equiv (8.0, 1.0)$

Figure 26: Image `orange` segmented with `mean-shift`. $(\sigma_s^2, \sigma_c^2) \equiv (8.0, 1.2)$

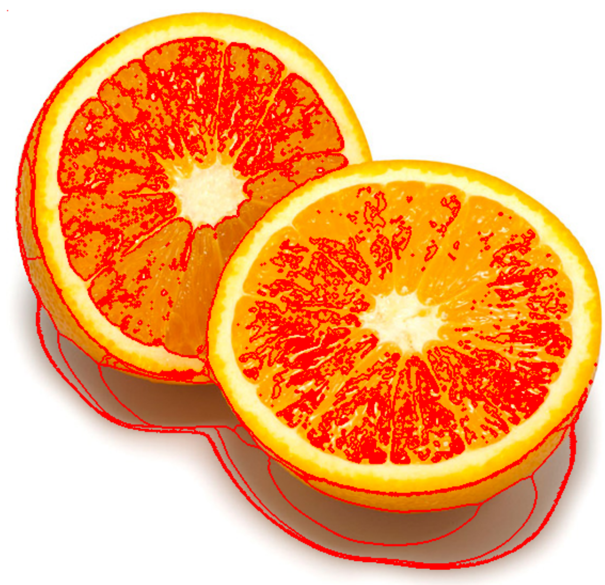

Figure 27: Segmentation bounds of image `orange`. $(\sigma_s^2, \sigma_c^2) \equiv (8.0, 1.2)$

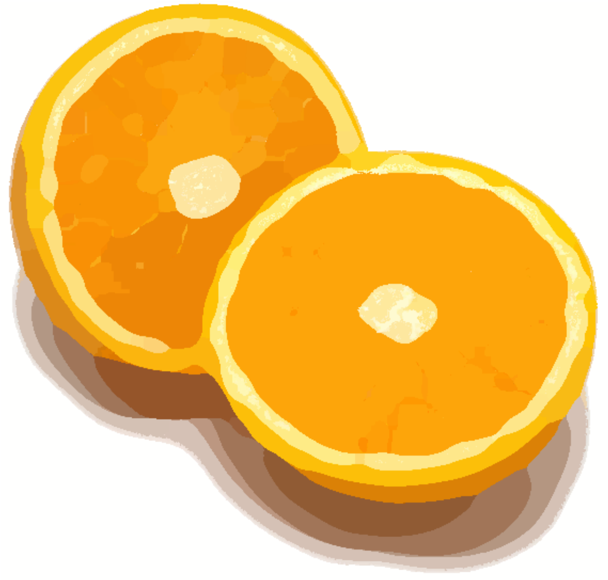

Figure 28: Image **orange** segmented with **mean-shift**. $(\sigma_s^2, \sigma_c^2) \equiv (8.0, 1.4)$

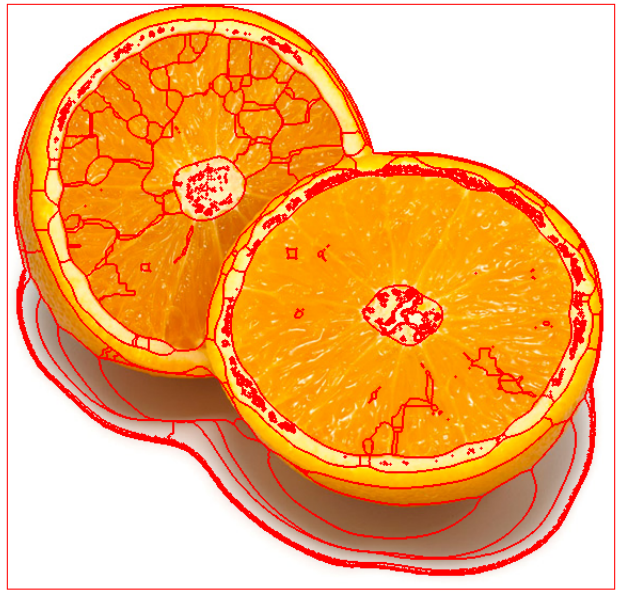

Figure 29: Segmentation bounds of image **orange**. $(\sigma_s^2, \sigma_c^2) \equiv (8.0, 1.4)$

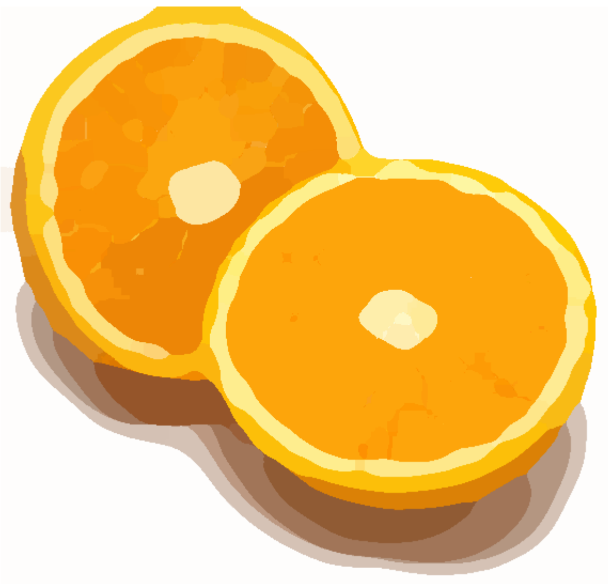

Figure 30: Image **orange** segmented with **mean-shift**. $(\sigma_s^2, \sigma_c^2) \equiv (8.0, 1.6)$

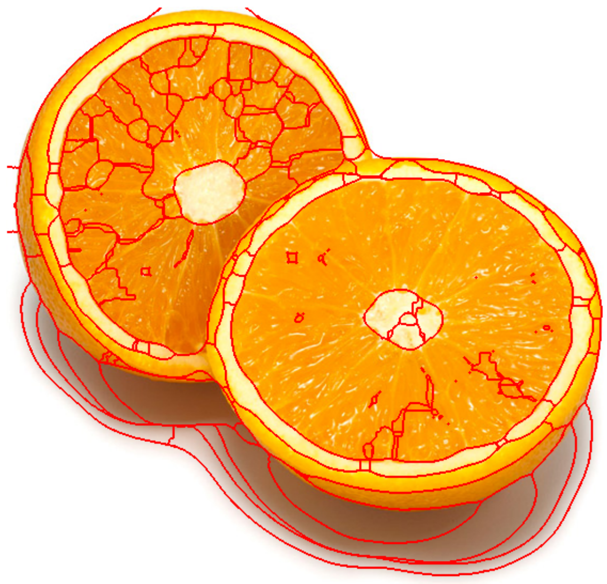

Figure 31: Segmentation bounds of image **orange**. $(\sigma_s^2, \sigma_c^2) \equiv (8.0, 1.6)$

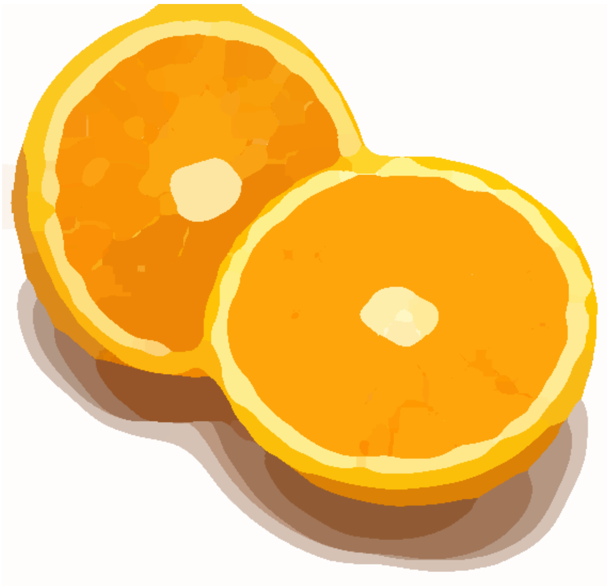

Figure 32: Image `orange` segmented with `mean-shift`. $(\sigma_s^2, \sigma_c^2) \equiv (8.0, 1.8)$

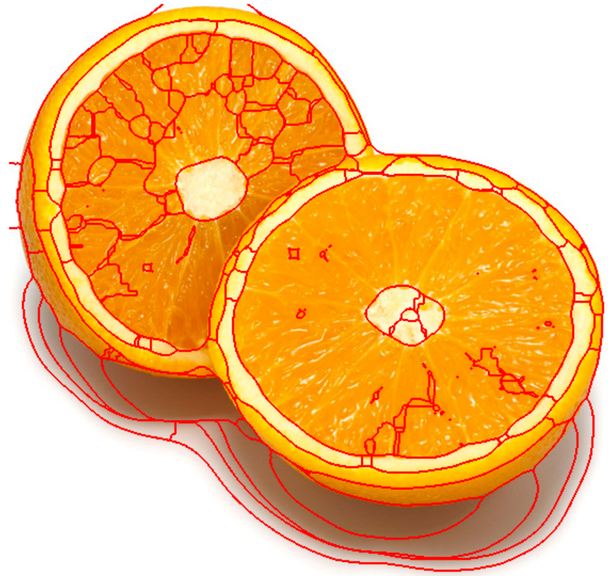

Figure 33: Segmentation bounds of image `orange`. $(\sigma_s^2, \sigma_c^2) \equiv (8.0, 1.8)$

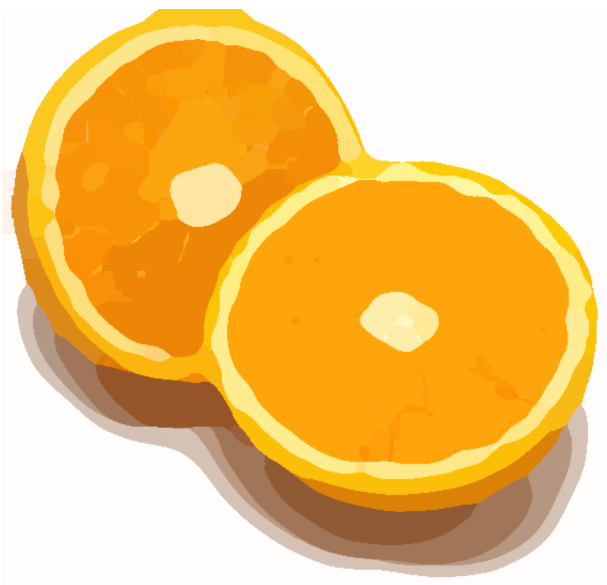

Figure 34: Image `orange` segmented with `mean-shift`. $(\sigma_s^2, \sigma_c^2) \equiv (8.0, 2.0)$

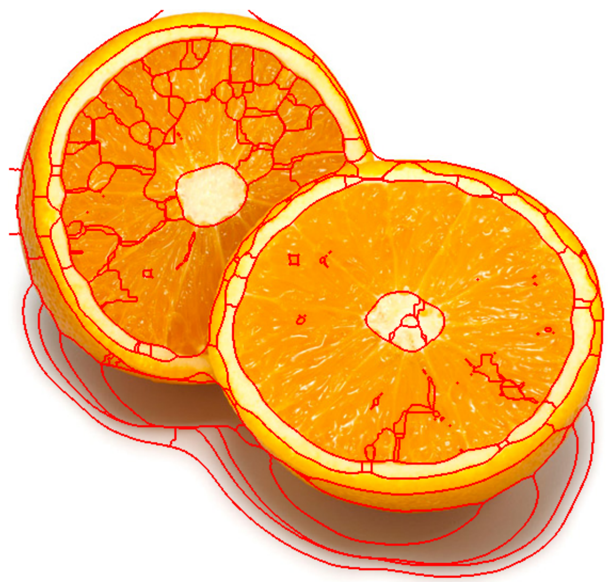

Figure 35: Segmentation bounds of image `orange`. $(\sigma_s^2, \sigma_c^2) \equiv (8.0, 2.0)$

2.1.3 Image **tiger1**. Varying σ_s^2.

Figure 36: Image `tiger1` segmented with `mean-shift`. $(\sigma_s^2, \sigma_c^2) \equiv (3.0, 4.0)$

Figure 37: Segmentation bounds of image `tiger1`. $(\sigma_s^2, \sigma_c^2) \equiv (3.0, 4.0)$

Figure 38: Image `tiger1` segmented with `mean-shift`. $(\sigma_s^2, \sigma_c^2) \equiv (4.0, 4.0)$

Figure 39: Segmentation bounds of image `tiger1`. $(\sigma_s^2, \sigma_c^2) \equiv (4.0, 4.0)$

Figure 40: Image `tiger1` segmented with `mean-shift`. $(\sigma_s^2, \sigma_c^2) \equiv (5.0, 4.0)$

Figure 41: Segmentation bounds of image `tiger1`. $(\sigma_s^2, \sigma_c^2) \equiv (5.0, 4.0)$

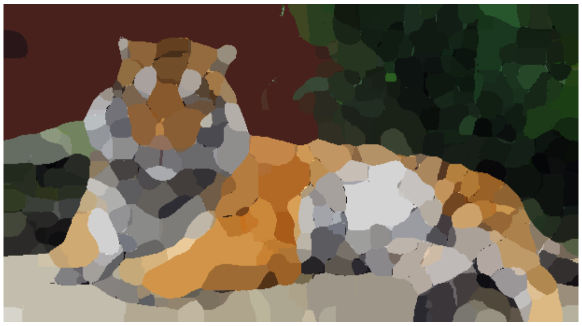

Figure 42: Image `tiger1` segmented with `mean-shift`. $(\sigma_s^2, \sigma_c^2) \equiv (8.0, 4.0)$

Figure 43: Segmentation bounds of image `tiger1`. $(\sigma_s^2, \sigma_c^2) \equiv (8.0, 4.0)$

Figure 44: Image `tiger1` segmented with `mean-shift`. $(\sigma_s^2, \sigma_c^2) \equiv (10.0, 4.0)$

Figure 45: Segmentation bounds of image `tiger1`. $(\sigma_s^2, \sigma_c^2) \equiv (10.0, 4.0)$

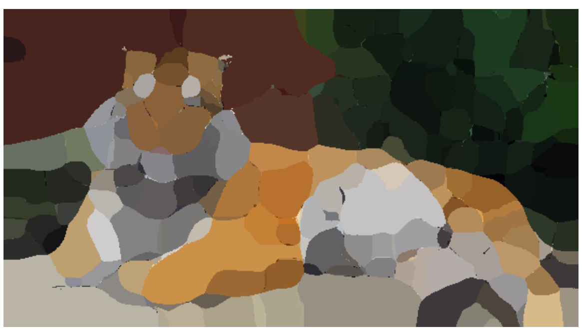

Figure 46: Image `tiger1` segmented with `mean-shift`. $(\sigma_s^2, \sigma_c^2) \equiv (12.0, 4.0)$

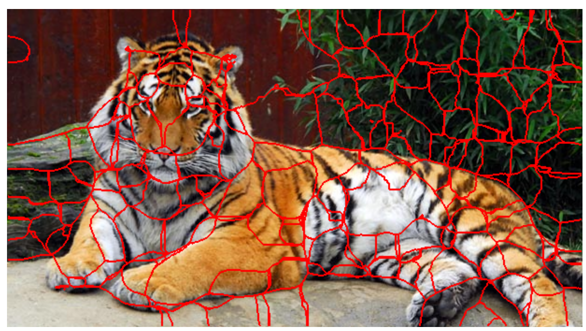

Figure 47: Segmentation bounds of image `tiger1`. $(\sigma_s^2, \sigma_c^2) \equiv (12.0, 4.0)$

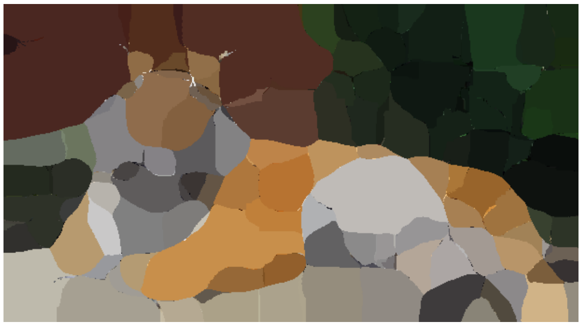

Figure 48: Image **tiger1** segmented with **mean-shift**. $(\sigma_s^2, \sigma_c^2) \equiv (15.0, 4.0)$

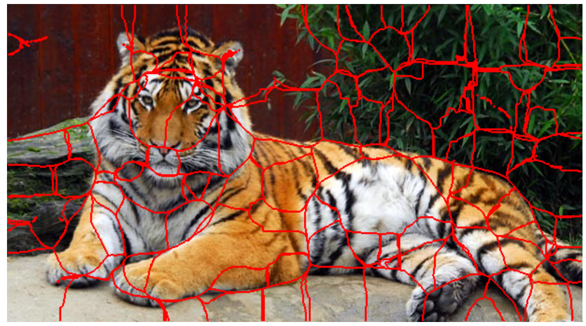

Figure 49: Segmentation bounds of image **tiger1**. $(\sigma_s^2, \sigma_c^2) \equiv (15.0, 4.0)$

Figure 50: Image **tiger1** segmented with **mean-shift**. $(\sigma_s^2, \sigma_c^2) \equiv (20.0, 4.0)$

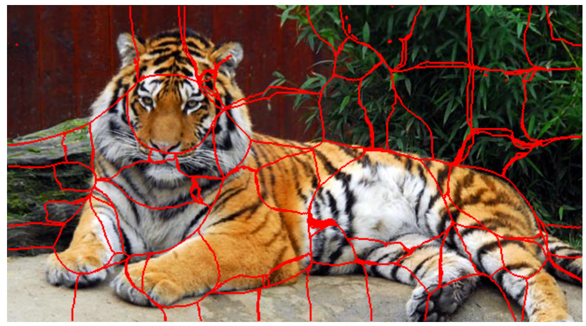

Figure 51: Segmentation bounds of image **tiger1**. $(\sigma_s^2, \sigma_c^2) \equiv (20.0, 4.0)$

2.1.4 Image **tiger1**. Varying σ_c^2.

Figure 52: Image **tiger1** segmented with **mean-shift**. $(\sigma_s^2, \sigma_c^2) \equiv (8.0, 1.0)$

Figure 53: Segmentation bounds of image **tiger1**. $(\sigma_s^2, \sigma_c^2) \equiv (8.0, 1.0)$

Figure 54: Image `tiger1` segmented with `mean-shift`. $(\sigma_s^2, \sigma_c^2) \equiv (8.0, 1.2)$

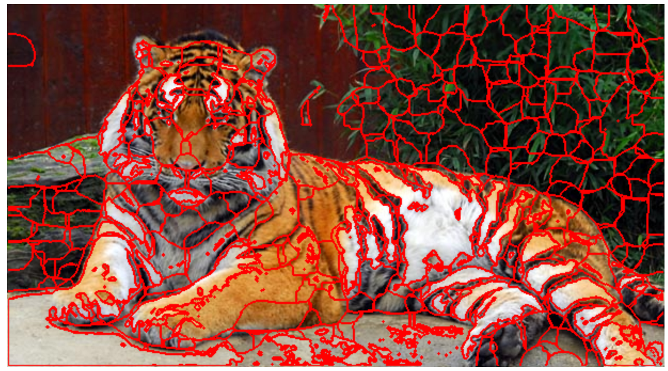

Figure 55: Segmentation bounds of image `tiger1`. $(\sigma_s^2, \sigma_c^2) \equiv (8.0, 1.2)$

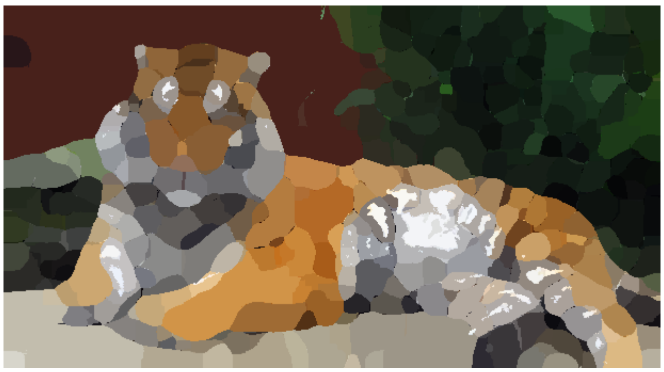

Figure 56: Image `tiger1` segmented with `mean-shift`. $(\sigma_s^2, \sigma_c^2) \equiv (8.0, 1.4)$

Figure 57: Segmentation bounds of image `tiger1`. $(\sigma_s^2, \sigma_c^2) \equiv (8.0, 1.4)$

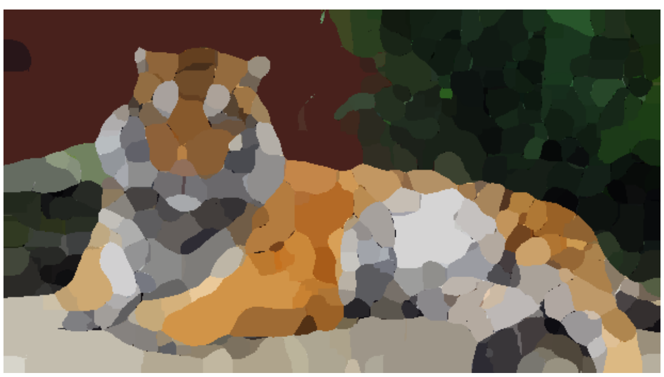

Figure 58: Image `tiger1` segmented with `mean-shift`. $(\sigma_s^2, \sigma_c^2) \equiv (8.0, 1.6)$

Figure 59: Segmentation bounds of image `tiger1`. $(\sigma_s^2, \sigma_c^2) \equiv (8.0, 1.6)$

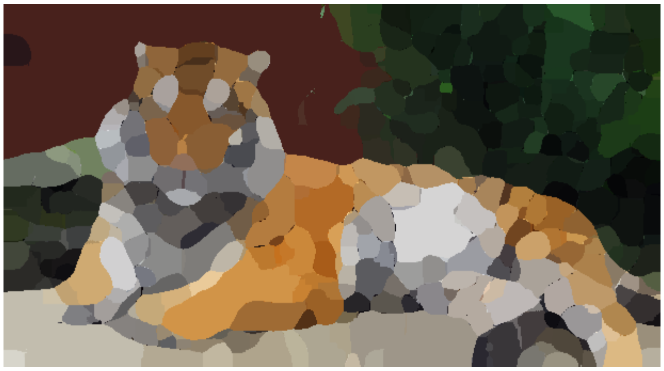

Figure 60: Image **tiger1** segmented with **mean-shift**. $(\sigma_s^2, \sigma_c^2) \equiv (8.0, 1.8)$

Figure 61: Segmentation bounds of image **tiger1**. $(\sigma_s^2, \sigma_c^2) \equiv (8.0, 1.8)$

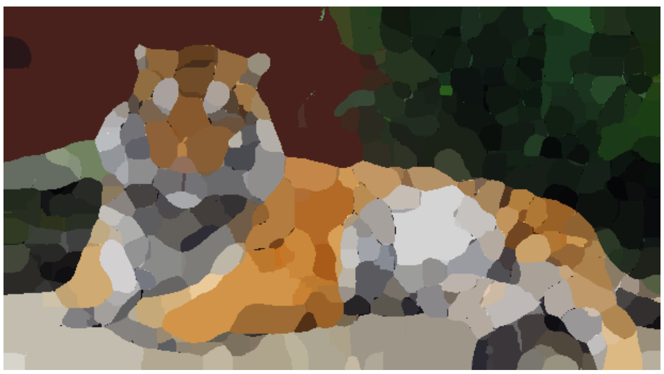

Figure 62: Image **tiger1** segmented with **mean-shift**. $(\sigma_s^2, \sigma_c^2) \equiv (8.0, 2.0)$

Figure 63: Segmentation bounds of image **tiger1**. $(\sigma_s^2, \sigma_c^2) \equiv (8.0, 2.0)$

2.1.5 Image **tiger2**. Varying σ_s^2.

Figure 64: Image **tiger2** segmented with **mean-shift**. $(\sigma_s^2, \sigma_c^2) \equiv (3.0, 4.0)$

Figure 65: Segmentation bounds of image **tiger2**. $(\sigma_s^2, \sigma_c^2) \equiv (3.0, 4.0)$

Figure 66: Image `tiger2` segmented with `mean-shift`. $(\sigma_s^2, \sigma_c^2) \equiv (4.0, 4.0)$

Figure 67: Segmentation bounds of image `tiger2`. $(\sigma_s^2, \sigma_c^2) \equiv (4.0, 4.0)$

Figure 68: Image `tiger2` segmented with `mean-shift`. $(\sigma_s^2, \sigma_c^2) \equiv (5.0, 4.0)$

Figure 69: Segmentation bounds of image `tiger2`. $(\sigma_s^2, \sigma_c^2) \equiv (5.0, 4.0)$

Figure 70: Image `tiger2` segmented with `mean-shift`. $(\sigma_s^2, \sigma_c^2) \equiv (8.0, 4.0)$

Figure 71: Segmentation bounds of image `tiger2`. $(\sigma_s^2, \sigma_c^2) \equiv (8.0, 4.0)$

Figure 72: Image `tiger2` segmented with `mean-shift`. $(\sigma_s^2, \sigma_c^2) \equiv (10.0, 4.0)$

Figure 73: Segmentation bounds of image `tiger2`. $(\sigma_s^2, \sigma_c^2) \equiv (10.0, 4.0)$

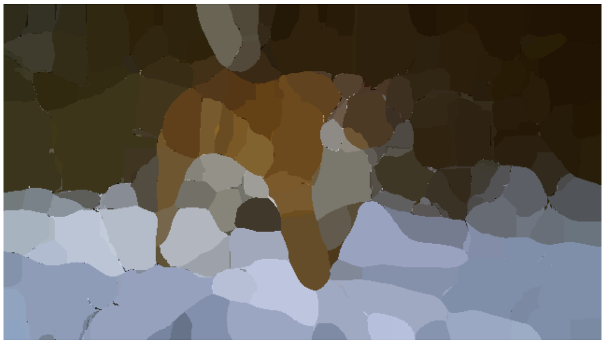

Figure 74: Image `tiger2` segmented with `mean-shift`. $(\sigma_s^2, \sigma_c^2) \equiv (12.0, 4.0)$

Figure 75: Segmentation bounds of image `tiger2`. $(\sigma_s^2, \sigma_c^2) \equiv (12.0, 4.0)$

Figure 76: Image `tiger2` segmented with `mean-shift`. $(\sigma_s^2, \sigma_c^2) \equiv (15.0, 4.0)$

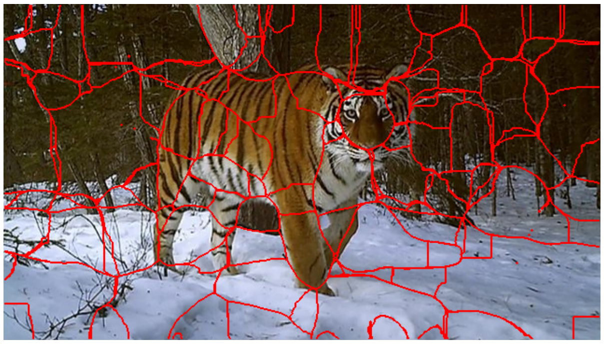

Figure 77: Segmentation bounds of image `tiger2`. $(\sigma_s^2, \sigma_c^2) \equiv (15.0, 4.0)$

Figure 78: Image **tiger2** segmented with **mean-shift**. $(\sigma_s^2, \sigma_c^2) \equiv (20.0, 4.0)$

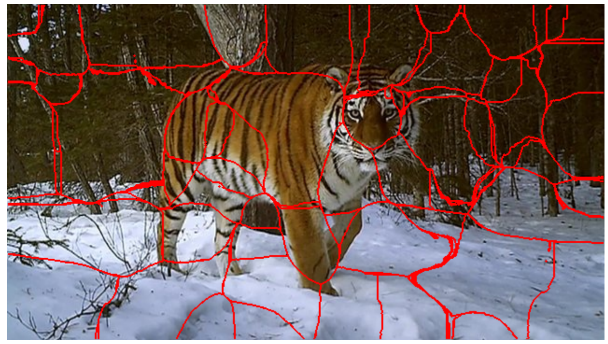

Figure 79: Segmentation bounds of image **tiger2**. $(\sigma_s^2, \sigma_c^2) \equiv (20.0, 4.0)$

2.1.6 Image **tiger2**. Varying σ_c^2.

Figure 80: Image **tiger2** segmented with **mean-shift**. $(\sigma_s^2, \sigma_c^2) \equiv (8.0, 1.0)$

Figure 81: Segmentation bounds of image **tiger2**. $(\sigma_s^2, \sigma_c^2) \equiv (8.0, 1.0)$

Figure 82: Image **tiger2** segmented with **mean-shift**. $(\sigma_s^2, \sigma_c^2) \equiv (8.0, 1.2)$

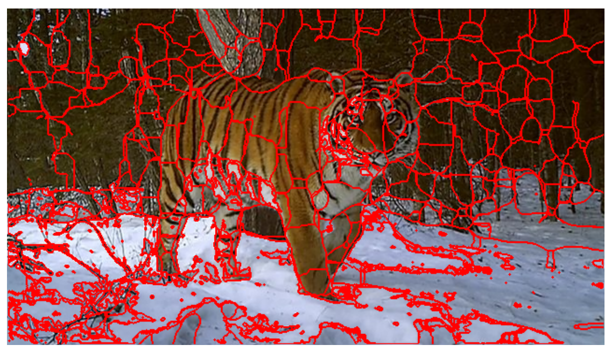

Figure 83: Segmentation bounds of image **tiger2**. $(\sigma_s^2, \sigma_c^2) \equiv (8.0, 1.2)$

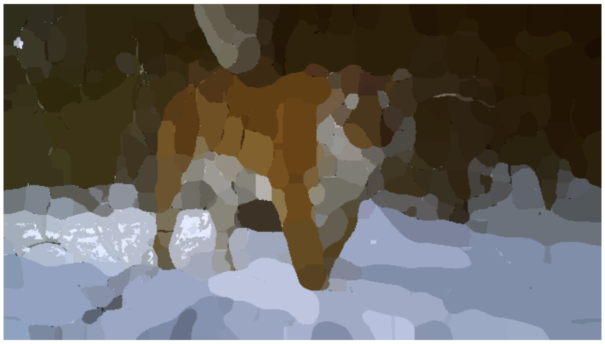

Figure 84: Image `tiger2` segmented with `mean-shift`. $(\sigma_s^2, \sigma_c^2) \equiv (8.0, 1.4)$

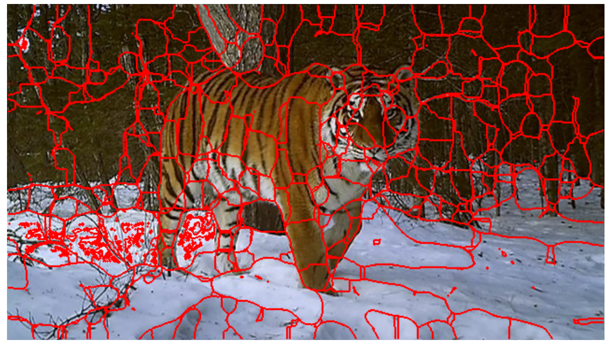

Figure 85: Segmentation bounds of image `tiger2`. $(\sigma_s^2, \sigma_c^2) \equiv (8.0, 1.4)$

Figure 86: Image `tiger2` segmented with `mean-shift`. $(\sigma_s^2, \sigma_c^2) \equiv (8.0, 1.6)$

Figure 87: Segmentation bounds of image `tiger2`. $(\sigma_s^2, \sigma_c^2) \equiv (8.0, 1.6)$

Figure 88: Image `tiger2` segmented with `mean-shift`. $(\sigma_s^2, \sigma_c^2) \equiv (8.0, 1.8)$

Figure 89: Segmentation bounds of image `tiger2`. $(\sigma_s^2, \sigma_c^2) \equiv (8.0, 1.8)$

Figure 90: Image **tiger2** segmented with **mean-shift**. $(\sigma_s^2, \sigma_c^2) \equiv (8.0, 2.0)$

Figure 91: Segmentation bounds of image **tiger2**. $(\sigma_s^2, \sigma_c^2) \equiv (8.0, 2.0)$

2.1.7 Image **tiger3**. Varying σ_s^2.

Figure 92: Image **tiger3** segmented with **mean-shift**. $(\sigma_s^2, \sigma_c^2) \equiv (3.0, 4.0)$

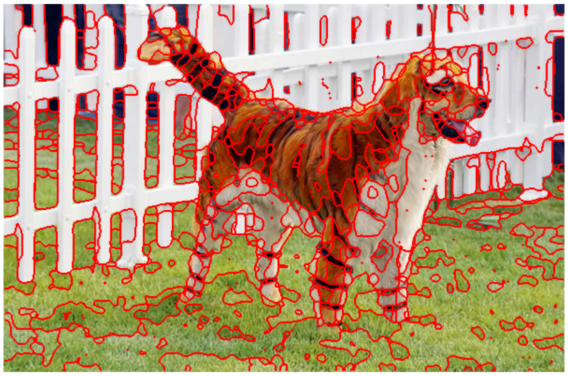

Figure 93: Segmentation bounds of image **tiger3**. $(\sigma_s^2, \sigma_c^2) \equiv (3.0, 4.0)$

Figure 94: Image **tiger3** segmented with **mean-shift**. $(\sigma_s^2, \sigma_c^2) \equiv (4.0, 4.0)$

Figure 95: Segmentation bounds of image **tiger3**. $(\sigma_s^2, \sigma_c^2) \equiv (4.0, 4.0)$

Figure 96: Image `tiger3` segmented with `mean-shift`. $(\sigma_s^2, \sigma_c^2) \equiv (5.0, 4.0)$

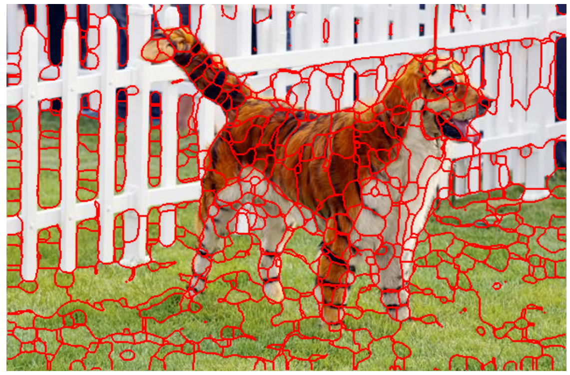

Figure 97: Segmentation bounds of image `tiger3`. $(\sigma_s^2, \sigma_c^2) \equiv (5.0, 4.0)$

Figure 98: Image `tiger3` segmented with `mean-shift`. $(\sigma_s^2, \sigma_c^2) \equiv (8.0, 4.0)$

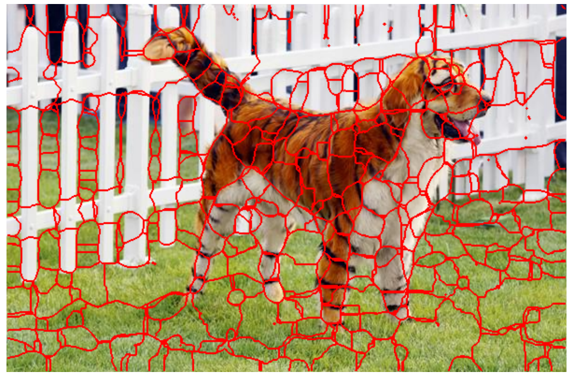

Figure 99: Segmentation bounds of image `tiger3`. $(\sigma_s^2, \sigma_c^2) \equiv (8.0, 4.0)$

Figure 100: Image `tiger3` segmented with `mean-shift`. $(\sigma_s^2, \sigma_c^2) \equiv (10.0, 4.0)$

Figure 101: Segmentation bounds of image `tiger3`. $(\sigma_s^2, \sigma_c^2) \equiv (10.0, 4.0)$

Figure 102: Image `tiger3` segmented with `mean-shift`. $(\sigma_s^2, \sigma_c^2) \equiv (12.0, 4.0)$

Figure 103: Segmentation bounds of image `tiger3`. $(\sigma_s^2, \sigma_c^2) \equiv (12.0, 4.0)$

Figure 104: Image `tiger3` segmented with `mean-shift`. $(\sigma_s^2, \sigma_c^2) \equiv (15.0, 4.0)$

Figure 105: Segmentation bounds of image `tiger3`. $(\sigma_s^2, \sigma_c^2) \equiv (15.0, 4.0)$

Figure 106: Image `tiger3` segmented with `mean-shift`. $(\sigma_s^2, \sigma_c^2) \equiv (20.0, 4.0)$

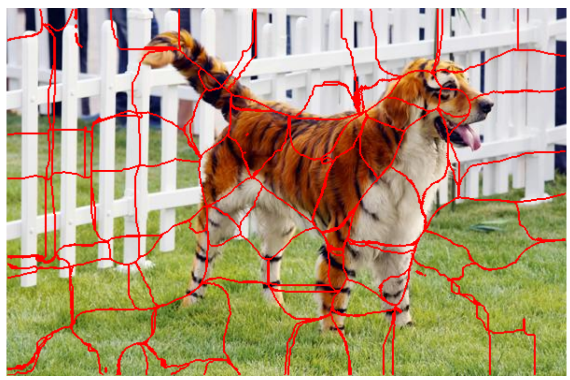

Figure 107: Segmentation bounds of image `tiger3`. $(\sigma_s^2, \sigma_c^2) \equiv (20.0, 4.0)$

2.1.8 Image **tiger3**. Varying σ_c^2.

Figure 108: Image `tiger3` segmented with `mean-shift`. $(\sigma_s^2, \sigma_c^2) \equiv (8.0, 1.0)$

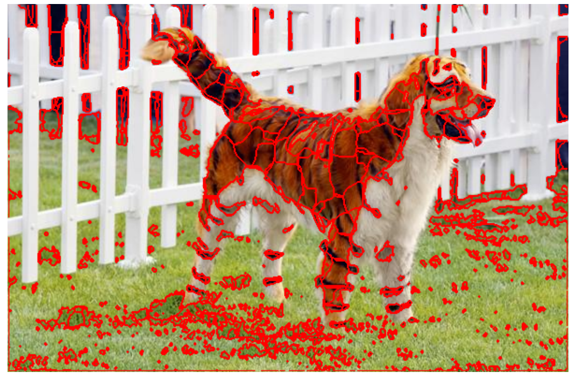

Figure 109: Segmentation bounds of image `tiger3`. $(\sigma_s^2, \sigma_c^2) \equiv (8.0, 1.0)$

Figure 110: Image `tiger3` segmented with `mean-shift`. $(\sigma_s^2, \sigma_c^2) \equiv (8.0, 1.2)$

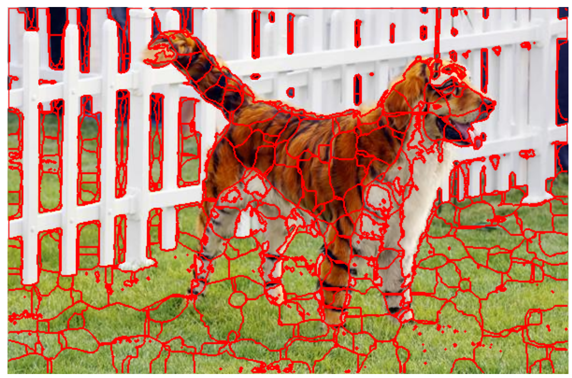

Figure 111: Segmentation bounds of image `tiger3`. $(\sigma_s^2, \sigma_c^2) \equiv (8.0, 1.2)$

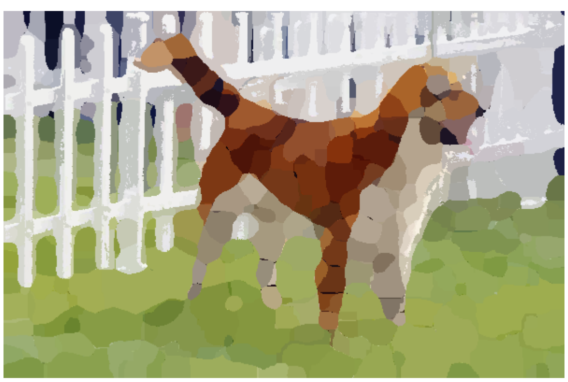

Figure 112: Image `tiger3` segmented with `mean-shift`. $(\sigma_s^2, \sigma_c^2) \equiv (8.0, 1.4)$

Figure 113: Segmentation bounds of image `tiger3`. $(\sigma_s^2, \sigma_c^2) \equiv (8.0, 1.4)$

Figure 114: Image `tiger3` segmented with `mean-shift`. $(\sigma_s^2, \sigma_c^2) \equiv (8.0, 1.6)$

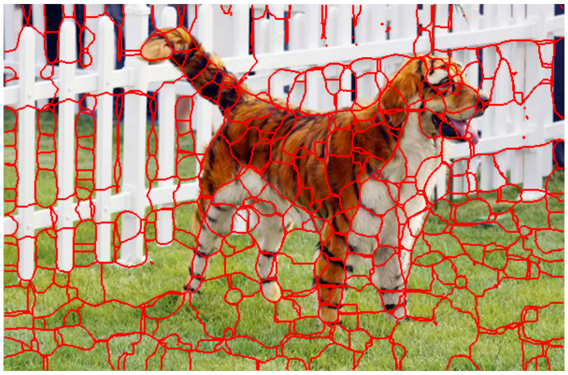

Figure 115: Segmentation bounds of image `tiger3`. $(\sigma_s^2, \sigma_c^2) \equiv (8.0, 1.6)$

Figure 116: Image `tiger3` segmented with `mean-shift`. $(\sigma_s^2, \sigma_c^2) \equiv (8.0, 1.8)$

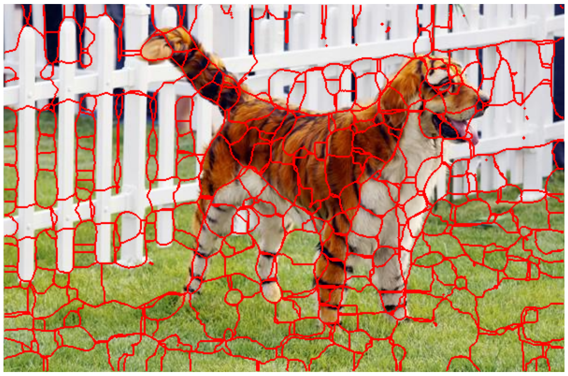

Figure 117: Segmentation bounds of image `tiger3`. $(\sigma_s^2, \sigma_c^2) \equiv (8.0, 1.8)$

Figure 118: Image `tiger3` segmented with `mean-shift`. $(\sigma_s^2, \sigma_c^2) \equiv (8.0, 2.0)$

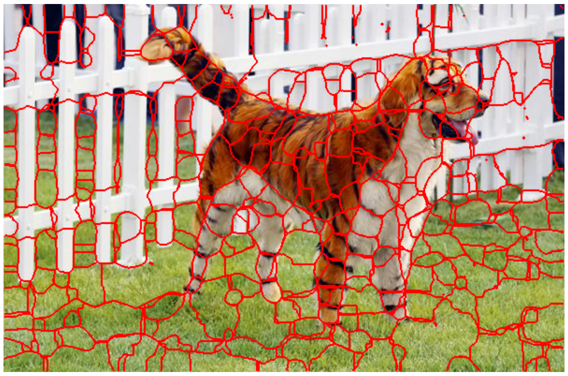

Figure 119: Segmentation bounds of image `tiger3`. $(\sigma_s^2, \sigma_c^2) \equiv (8.0, 2.0)$

2.2 Best results

Having as purpose to segment the 4 images in as large segments as possible, without them covering more than one object in each one, the configuration that was used is illustrated in table 3.

image	σ_s^2	σ_c^2
`orange`	3.0	4.0
`tiger1`	8.0	4.0
`tiger2`	10.0	4.0
`tiger3`	8.0	4.0

Table 3: Spatial and colour bandwidths that result in good segmentation by the above purpose per image.

Figures 120 - 127 illustrate these results.

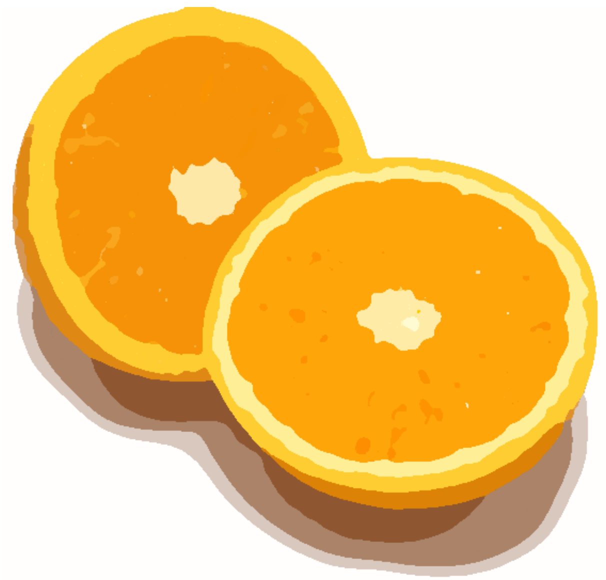

Figure 120: Image `orange` segmented with `mean-shift`. $(\sigma_s^2, \sigma_c^2) \equiv (3.0, 4.0)$

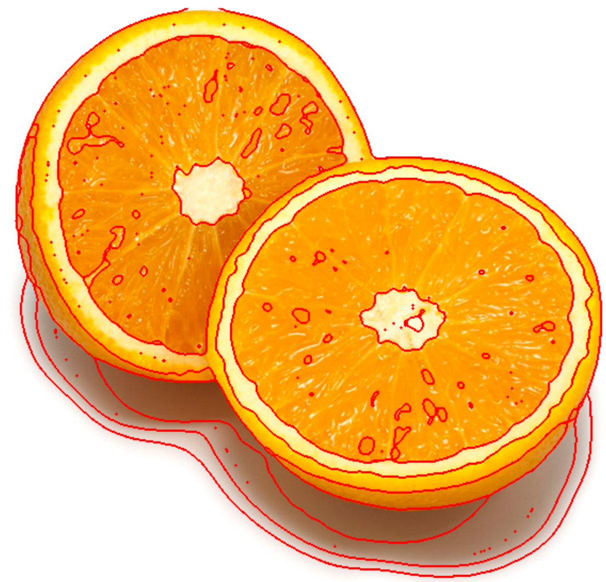

Figure 121: Segmentation bounds of image `orange`. $(\sigma_s^2, \sigma_c^2) \equiv (3.0, 4.0)$

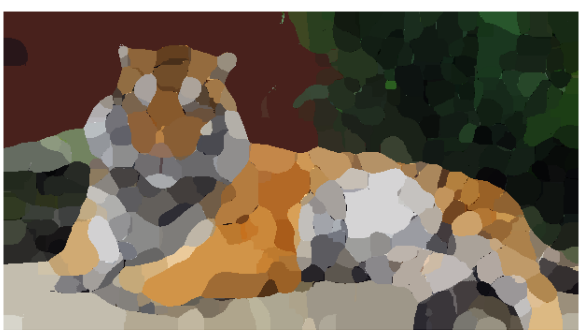

Figure 122: Image `tiger1` segmented with `mean-shift`. $(\sigma_s^2, \sigma_c^2) \equiv (8.0, 4.0)$

Figure 123: Segmentation bounds of image `tiger1`. $(\sigma_s^2, \sigma_c^2) \equiv (8.0, 4.0)$

Figure 124: Image `tiger2` segmented with `mean-shift`. $(\sigma_s^2, \sigma_c^2) \equiv (10.0, 4.0)$

Figure 125: Segmentation bounds of image `tiger2`. $(\sigma_s^2, \sigma_c^2) \equiv (10.0, 4.0)$

Figure 126: Image `tiger3` segmented with `mean-shift`. $(\sigma_s^2, \sigma_c^2) \equiv (8.0, 4.0)$

Figure 127: Segmentation bounds of image `tiger3`. $(\sigma_s^2, \sigma_c^2) \equiv (8.0, 4.0)$

2.3 Question 5

Figures 8 - 23, 36 - 51, 64 - 79 and 92 - 107 illustrate the `mean-shift` segmentation method for varying values of spatial bandwidths, and colour bandwidth set to $\sigma_c^2 = 4.0$.

As seen in the above figures, the value of the spatial bandwidth is proportional to the area of the regions of interest, that is, the resulting segments. As the area of the segments becomes larger, their number becomes smaller, and thus so does the number of modes. Hence, there is a coarser approximation both spatial-wise and colour-wise. Another way to look at this is that the bandwidth σ_s^2 controls the width of the density function around each image point. The smaller it is, the clearer the peaks of the density function in `5D` space are, hence the more distinct the gradient ascent is, the more representative of the actual coherence the modes are, and the better the assignment accuracy. However, as the spatial bandwidth increases, the width of the gaussians increases, and gaussians become more spread-out into space, while blending together for neighbouring pixels, and, hence, making it more easy for pixels to be assigned to a more spread-out mode.

Figures 24 - 35, 52 - 63, 80 - 91 and 108 - 119 illustrate the `mean-shift` segmentation method for varying values of colour bandwidths, and spatial bandwidth set to $\sigma_s^2 = 8.0$.

As seen in the above figures, a higher value for the colour bandwidth results in a better colour approximation.

2.4 Question 6

K-means and mean-shift both treat the colour and/or position of pixels as samples from a probability distribution and try to determine its clusters or modes.

Unlike k-means, mean-shift cannot segment an image into a predefined number of segments. Furthermore k-means does not take the spatial dimension into account when segmenting; only the colour component. This means that a segment, which is only specified by its colour, can span over multiple regions, while this cannot be the case with mean-shift, whose segments are spatially concentrated.

3 Normalized Cut

3.1 Question 7

The ideal parameter setting for each of these images depend on various factors, along with the degree of segmentation that is desired.

For example, parameter `min_area` depends on the size of the image, the size of the objects that are to be segmented and the complexity of each object's structure. Figures 128 - 131 illustrate the effect that the variation of the value of `min_area` has on image `orange`. Notice that when `min_area` ≥ 100 the center of the left half of the orange cannot be found and is taken over by its surroundings.

Figure 128: Image `orange` segmented with the default settings and `min_area = 500`

Figure 129: Image `orange` segmented with the default settings and `min_area = 200`

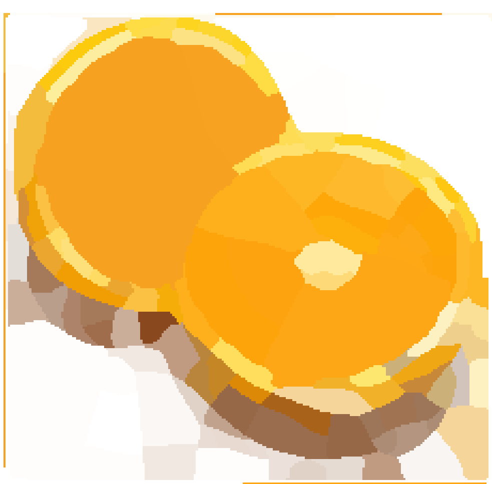

Figure 130: Image `orange` segmented with the default settings and `min_area = 100`

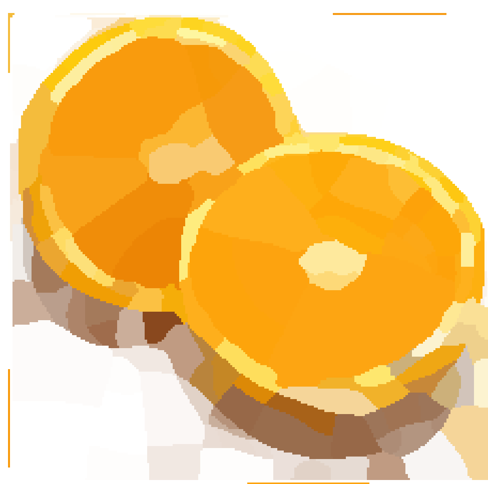

Figure 131: Image `orange` segmented with the default settings and `min_area = 10`

Parameter `ncut_thresh` depends on the degree of derisable segmentation, the spatial complexity and the diversity in colour of each image. Figures 132 - 137 illustrate the effect that the increase in the maximum allowed value for a cut to made has on image `tiger1`. This image has significantly more colour diversity than image `orange`.

Figure 132: Image `tiger1` segmented with the default settings and `ncuts_thresh = 0.01`

Figure 133: Image `tiger1` segmented with the default settings and `ncuts_thresh = 0.02`

Figure 134: Image `tiger1` segmented with the default settings and `ncuts_thresh = 0.05`

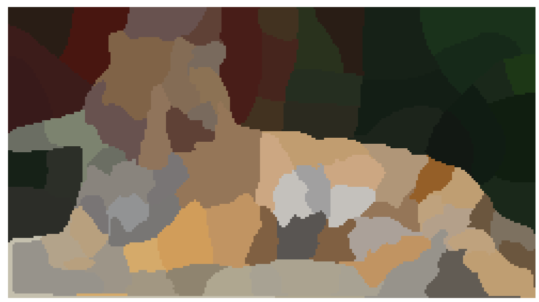

Figure 135: Image `tiger1` segmented with the default settings and `ncuts_thresh = 0.1`

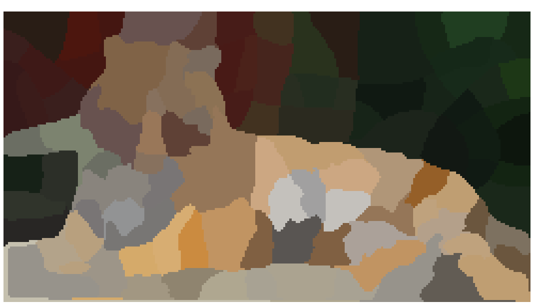

Figure 136: Image `tiger1` segmented with the default settings and `ncuts_thresh = 0.2`

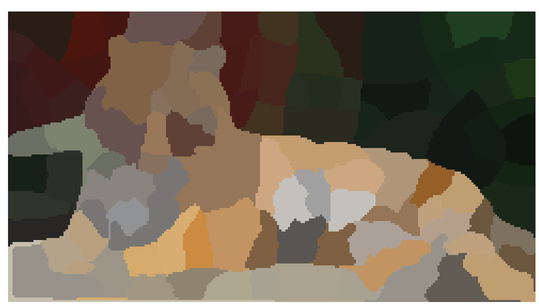

Figure 137: Image `tiger1` segmented with the default settings and `ncuts_thresh = 0.5`

Figures 138 - 143 illustrate the effect that the increase in the recursion depth has on image `tiger3`. Parameter `max_depth`.

Figure 138: Image `tiger3` segmented with the default settings and `max_depth = 1`

Figure 139: Image `tiger3` segmented with the default settings and `max_depth = 2`

Figure 140: Image `tiger3` segmented with the default settings and `max_depth = 4`

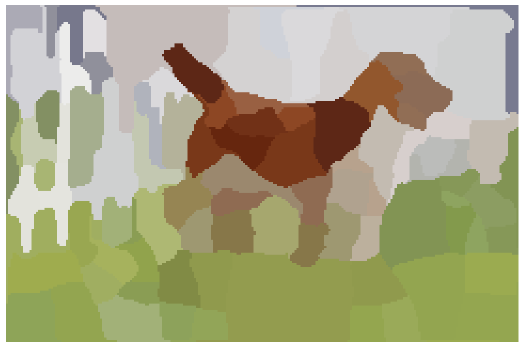

Figure 141: Image `tiger3` segmented with the default settings and `max_depth = 8`

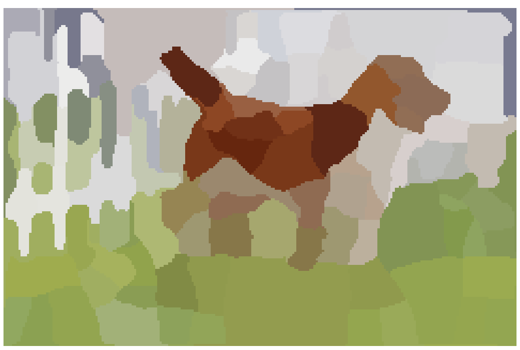

Figure 142: Image `tiger3` segmented with the default settings and `max_depth = 10`

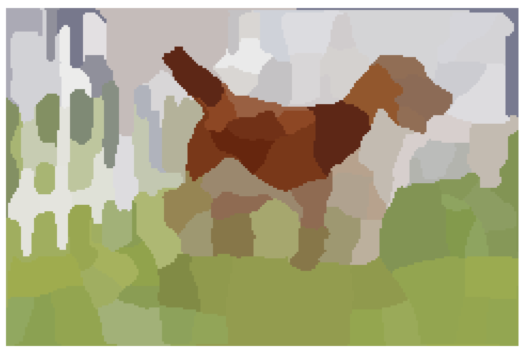

Figure 143: Image `tiger3` segmented with the default settings and `max_depth = 16`

The combination of the three parameters `(min_area, ncuts_thresh, max_depth)` will be different for the 4 images to the extent of the different attributes of each image. For instance figures 144 - 147 show that not only `min_area` has to be small (~ 10) in order for the center of the left-half of the orange to be depicted, but also higher values are needed for `ncuts_thresh` and `max_depth` in order for a more accurate segmentation to take place. The latter can be also said for images `tiger{1,2,3}`, since their colour and spatial diversity is higher than those of image `orange`, and more cuts need to be made in order to depict the little details in the depicted animal's skin.

Figures 144 - 159 illustrate the most reasonably well segmented results of applying the Normal Cut segmentation method to images `orange`, `tiger{1,2,3}`.

3.1.1 Best results - image `orange`

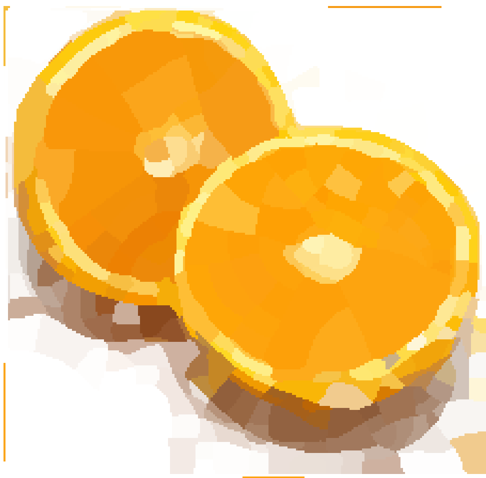

Figure 144: Image `orange` segmented with `(min_area, ncut_thresh, max_depth)` ≡ `(10, 0.2, 10)`.

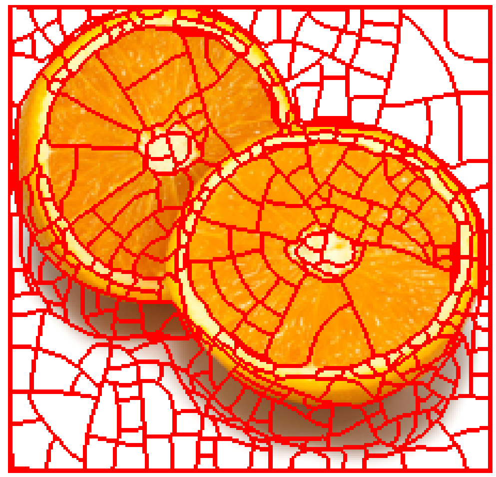

Figure 145: Image `orange` and the bounds of its segments. `(min_area, ncut_thresh, max_depth)` ≡ `(10, 0.2, 10)`.

Figure 146: Image `orange` segmented with `(min_area, ncut_thresh, max_depth)` ≡ `(10, 0.5, 16)`.

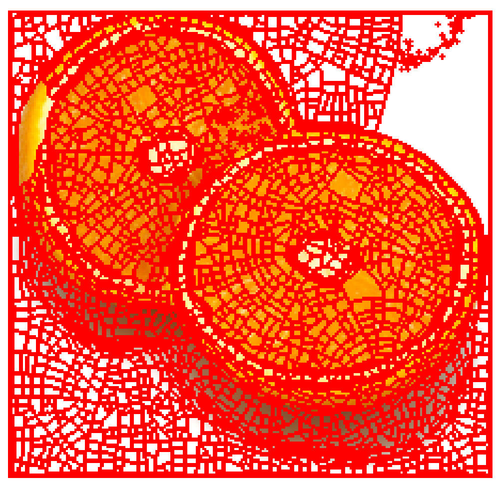

Figure 147: Image `orange` and the bounds of its segments. `(min_area, ncut_thresh, max_depth)` ≡ `(10, 0.5, 16)`.

3.1.2 Image **tiger1**

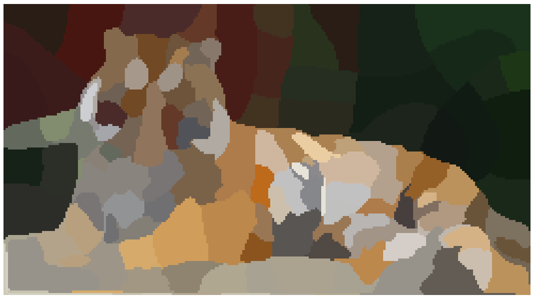

Figure 148: Image `tiger1` segmented with `(min_area, ncut_thresh, max_depth)` ≡ `(10, 0.1, 8)`.

Figure 149: Image `tiger1` and the bounds of its segments. `(min_area, ncut_thresh, max_depth)` ≡ `(10, 0.1, 8)`.

Figure 150: Image `tiger1` segmented with `(min_area, ncut_thresh, max_depth)` ≡ `(10, 0.2, 16)`.

Figure 151: Image `tiger1` and the bounds of its segments. `(min_area, ncut_thresh, max_depth)` ≡ `(10, 0.2, 16)`.

3.1.3 Image **tiger2**

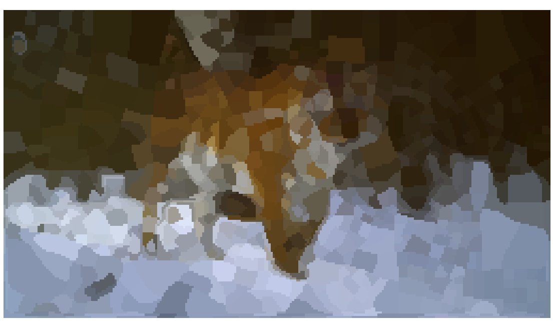

Figure 152: Image `tiger2` segmented with `(min_area, ncut_thresh, max_depth)` ≡ `(10, 0.5, 10)`.

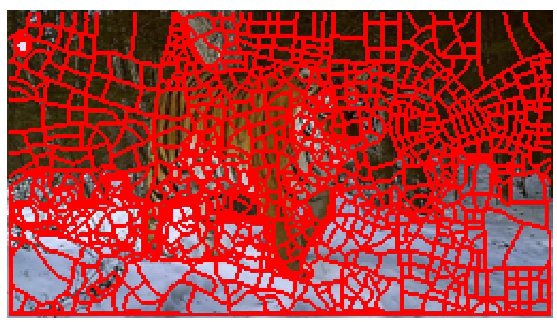

Figure 153: Image `tiger2` and the bounds of its segments. `(min_area, ncut_thresh, max_depth)` ≡ `(10, 0.5, 10)`.

Figure 154: Image `tiger2` segmented with `(min_area, ncut_thresh, max_depth)` ≡ `(10, 0.5, 16)`.

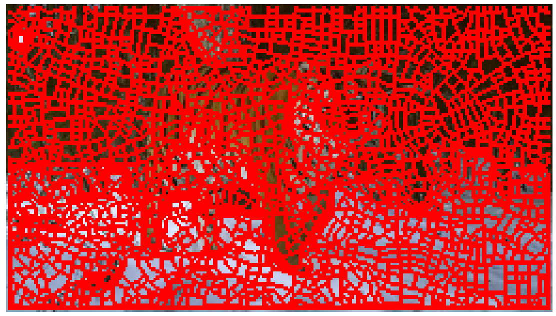

Figure 155: Image `tiger2` and the bounds of its segments. `(min_area, ncut_thresh, max_depth)` ≡ `(10, 0.5, 16)`.

3.1.4 Image **tiger3**

Figure 156: Image `tiger3` segmented with `(min_area, ncut_thresh, max_depth)` ≡ `(100, 0.1, 8)`.

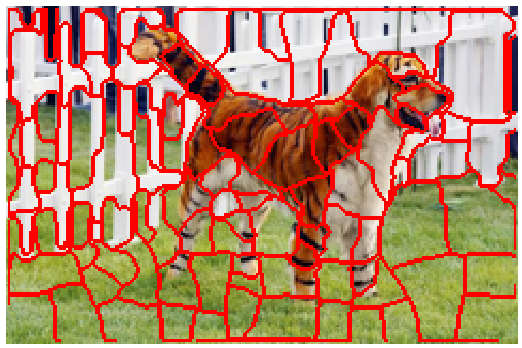

Figure 157: Image `tiger3` and the bounds of its segments. `(min_area, ncut_thresh, max_depth)` ≡ `(100, 0.1, 8)`.

Figure 158: Image `tiger3` segmented with `(min_area, ncut_thresh, max_depth)` ≡ `(10, 0.5, 10)`.

Figure 159: Image `tiger3` and the bounds of its segments. `(min_area, ncut_thresh, max_depth)` ≡ `(10, 0.5, 10)`.

3.2 Question 8

The two parameters that were able to reduce the number of segments while at the same time keeping a reasonable segmentation accuracy were both the `ncuts_thresh` and `max_depth` parameters as it can be seen in figures 132 - 143. Furthermore, parameter `min_area` can also affect the subdivision when increased, although not in the same capacity as the former two parameters.

3.3 Question 9

Since

$$Ncut(A,B) = cut(A,B)\Big(\frac{1}{assoc(A,V)} + \frac{1}{assoc(B,V)}\Big) \tag{1}$$

if $assoc(V)$ represents the total number of edges in the graph, then

$$assoc(V) = assoc(A,V) + assoc(B,V) - cut(A,B) \tag{2}$$

since $cut(A,B)$ is included once in each $assoc(*,V)$. Equation 1 is then formed as

$$Ncut(A,B) = cut(A,B)\Big(\frac{1}{assoc(A,V)} + \frac{1}{assoc(V) - cut(A,B) - assoc(A,V)}\Big) \tag{3}$$

In equation 3, we consider all variables except $assoc(A,V)$ to be constants. Then, minimizing equation 3 means finding the value of $assoc(A,V)$ that satisfies equation 4:

$$\frac{d}{dassoc(A,V)} Ncut(A,B) = 0 \tag{4}$$

The derivative of $Ncut(A,B)$ with respect to $assoc(A,V)$, $\frac{d}{dassoc(A,V)}Ncut(A,B)$ is

$$\begin{gathered}\frac{d}{dassoc(A,V)} Ncut(A,B) = -\frac{cut(A,B)}{assoc(A,V)^2} + \frac{cut(A,B)}{(assoc(V) + cut(A,B) - assoc(A,V))^2} = \\ \frac{cut(A,B)(assoc(A,V)^2 - (assoc(A,V) - assoc(V) - cut(A,B))^2)}{(assoc(A,V)(assoc(A,V) - assoc(V) - cut(A,B))^2)} = \\ \frac{cut(A,B)(-2assoc(A,V)(assoc(V) + cut(A,B)) + (assoc(V) + cut(A,B))^2)}{(assoc(A,V)(assoc(A,V) - assoc(V) - cut(A,B))^2)} = \\ \frac{cut(A,B)(assoc(V) + cut(A,B))(-2assoc(A,V) + (assoc(V) + cut(A,B)))}{(assoc(A,V)(assoc(A,V) - assoc(V) - cut(A,B))^2)}\end{gathered} \tag{5}$$

Hence, if equation 4 is to be satisfied

$$assoc(A,V) = \frac{assoc(V) + cut(A,B)}{2} \tag{6}$$

and from equation 2

$$assoc(B, V) = \frac{assoc(V) + cut(A, B)}{2} = assoc(A, V) \tag{7}$$

Hence, the Normalized Cuts method, in theory at least, tries to minimize the cut by finding equally sized cuts. However, in practice this was not witnessed in general, as the problem of finding the optimal cut is a NP-complete problem and the methods that try to solve it and are at our disposal are approximative.

3.4 Question 10

Figures 160 - 191 compare the figures of sections 3.1.1, 3.1.2, 3.1.3 and 3.1.4 which were generated with `radius = 3` with their respective counterparts of `radius = 6`.

What is apparent from this comparison is that the higher the value of `radius`, that is, the larger the considered neighbourhood of a pixel, the lower the number of segments becomes. However, that does not mean that the segmentation becomes coarser; the case is quite the opposite. While a smaller radius produces an overfragmentation even for areas whose internal structure is reasonably homogeneous, a larger radius can split regions in finer detail along the edges of colour-wise heterogeneous regions and treat reasonably homogeneous regions in a more relaxed way, assigning them a single colour, instead of multiple hues of the same colour. This is quite evident in the segmentation of the centers of the two orange halves, the bush and wall over the tiger in image `tiger1`, the region of the white fur of the tiger in the same image, the distinction of the eyes of the tiger in image `tiger2` and the foreground and background of image `tiger3`.

3.4.1 Image `orange`

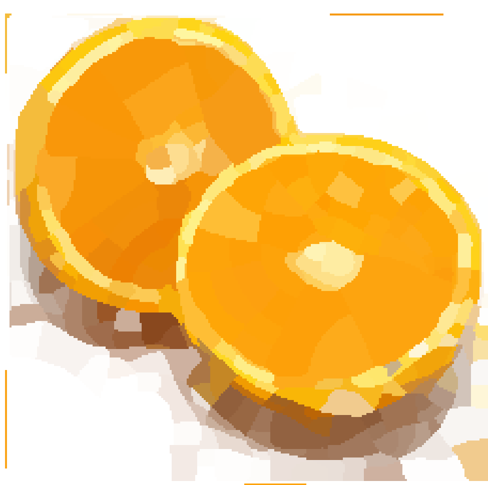

Figure 160: Image `orange` segmented with `(min_area, ncut_thresh, max_depth)` $\equiv$ `(10, 0.2, 10). radius = 3`

Figure 161: Image `orange` and the bounds of its segments. `(min_area, ncut_thresh, max_depth)` $\equiv$ `(10, 0.2, 10). radius = 6`

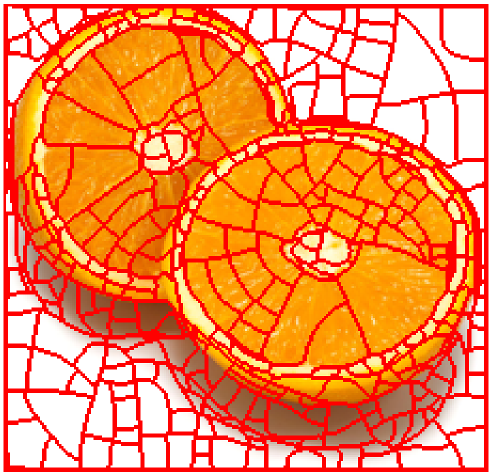

Figure 162: Image `orange` segmented with `(min_area, ncut_thresh, max_depth)` $\equiv$ `(10, 0.2, 10). radius = 3`

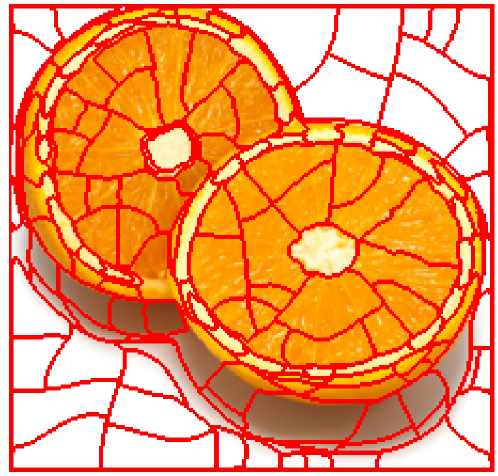

Figure 163: Image `orange` and the bounds of its segments. `(min_area, ncut_thresh, max_depth)` $\equiv$ `(10, 0.2, 10). radius = 6`

Figure 164: Image `orange` segmented with `(min_area, ncut_thresh, max_depth)` $\equiv$ `(10, 0.5, 16). radius = 3`

Figure 165: Image `orange` and the bounds of its segments. `(min_area, ncut_thresh, max_depth)` $\equiv$ `(10, 0.5, 16). radius = 6`

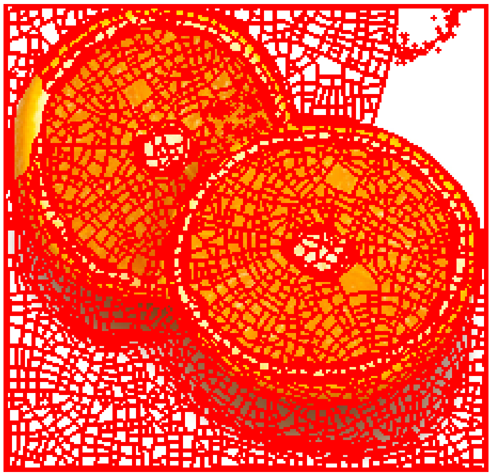

Figure 166: Image `orange` segmented with `(min_area, ncut_thresh, max_depth)` $\equiv$ `(10, 0.5, 16). radius = 3`

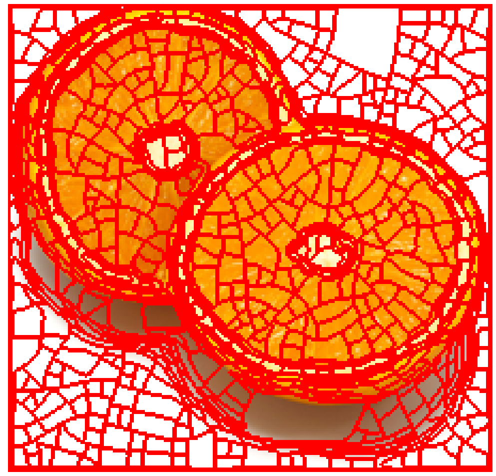

Figure 167: Image `orange` and the bounds of its segments. `(min_area, ncut_thresh, max_depth)` $\equiv$ `(10, 0.5, 16). radius = 6`

3.4.2 Image `tiger1`

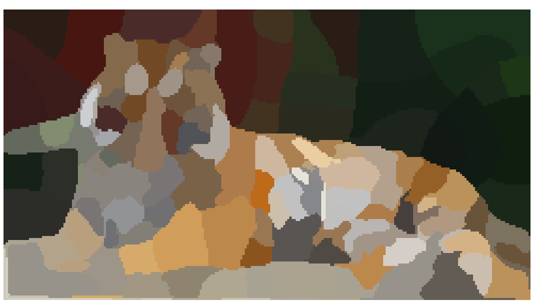

Figure 168: Image `tiger1` segmented with `(min_area, ncut_thresh, max_depth)` $\equiv$ `(10, 0.1, 8). radius = 3`

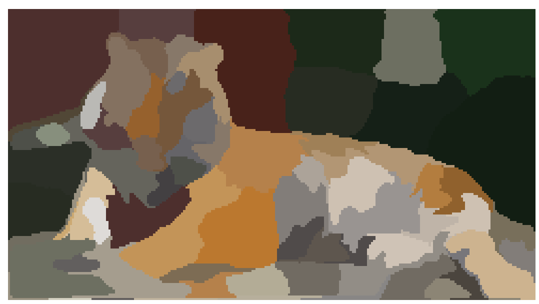

Figure 169: Image `tiger1` and the bounds of its segments. `(min_area, ncut_thresh, max_depth)` $\equiv$ `(10, 0.1, 8). radius = 6`

Figure 170: Image `tiger1` segmented with `(min_area, ncut_thresh, max_depth)` ≡ `(10, 0.1, 8)`. `radius = 3`

Figure 171: Image `tiger1` and the bounds of its segments. `(min_area, ncut_thresh, max_depth)` ≡ `(10, 0.1, 8)`. `radius = 6`

Figure 172: Image `tiger1` segmented with `(min_area, ncut_thresh, max_depth)` ≡ `(10, 0.2, 16)`. `radius = 3`

Figure 173: Image `tiger1` and the bounds of its segments. `(min_area, ncut_thresh, max_depth)` ≡ `(10, 0.2, 16)`. `radius = 6`

Figure 174: Image `tiger1` segmented with `(min_area, ncut_thresh, max_depth)` ≡ `(10, 0.2, 16)`. `radius = 3`

Figure 175: Image `tiger1` and the bounds of its segments. `(min_area, ncut_thresh, max_depth)` ≡ `(10, 0.2, 16)`. `radius = 6`

3.4.3 Image **tiger2**

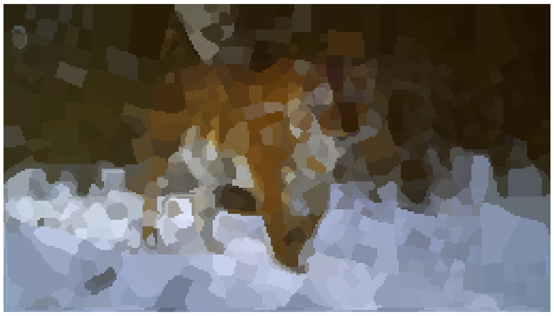

Figure 176: Image `tiger2` segmented with `(min_area, ncut_thresh, max_depth)` ≡ `(10, 0.5, 10)`. `radius = 3`

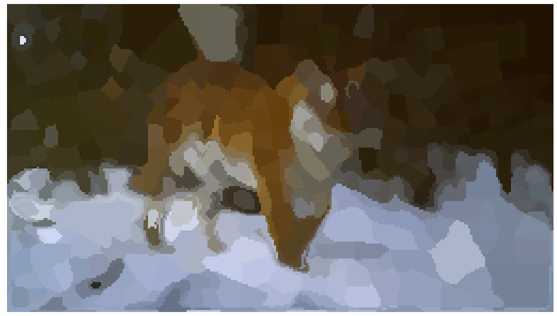

Figure 177: Image `tiger2` and the bounds of its segments. `(min_area, ncut_thresh, max_depth)` ≡ `(10, 0.5, 10)`. `radius = 6`

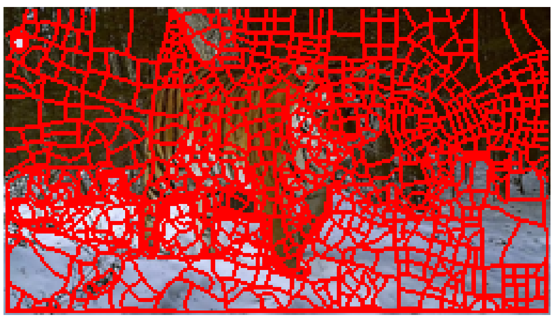

Figure 178: Image `tiger2` segmented with `(min_area, ncut_thresh, max_depth)` ≡ `(10, 0.5, 10)`. `radius = 3`

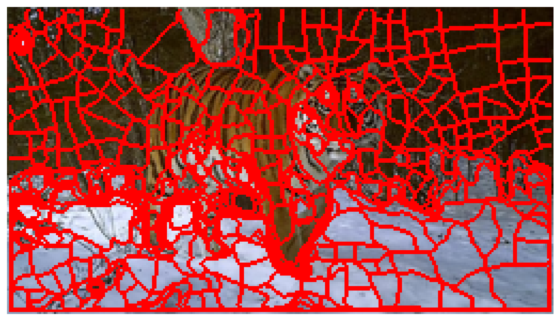

Figure 179: Image `tiger2` and the bounds of its segments. `(min_area, ncut_thresh, max_depth)` ≡ `(10, 0.5, 10)`. `radius = 6`

Figure 180: Image `tiger2` segmented with `(min_area, ncut_thresh, max_depth)` ≡ `(10, 0.5, 16)`. `radius = 3`

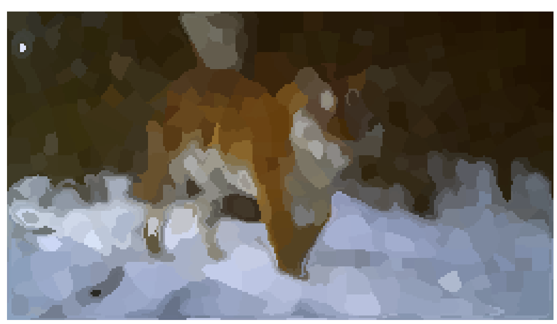

Figure 181: Image `tiger2` and the bounds of its segments. `(min_area, ncut_thresh, max_depth)` ≡ `(10, 0.5, 16)`. `radius = 6`

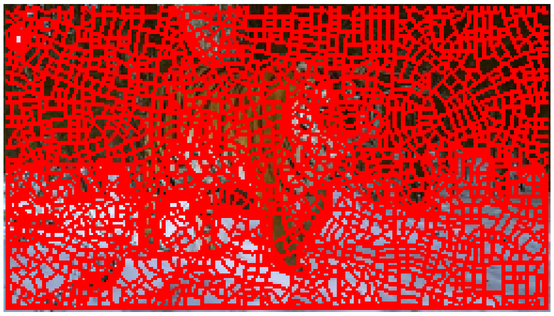

Figure 182: Image **tiger2** segmented with `(min_area, ncut_thresh, max_depth)` ≡ `(10, 0.5, 16). radius = 3`

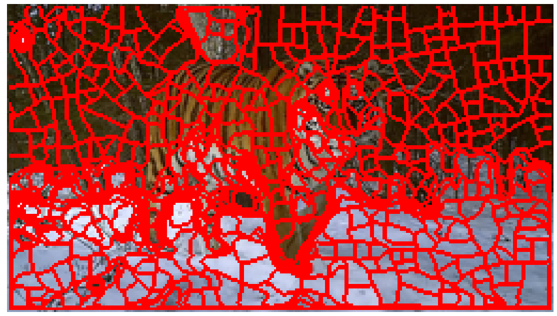

Figure 183: Image **tiger2** and the bounds of its segments. `(min_area, ncut_thresh, max_depth)` ≡ `(10, 0.5, 16). radius = 6`

3.4.4 Image **tiger3**

Figure 184: Image **tiger3** segmented with `(min_area, ncut_thresh, max_depth)` ≡ `(10, 0.5, 10). radius = 3`

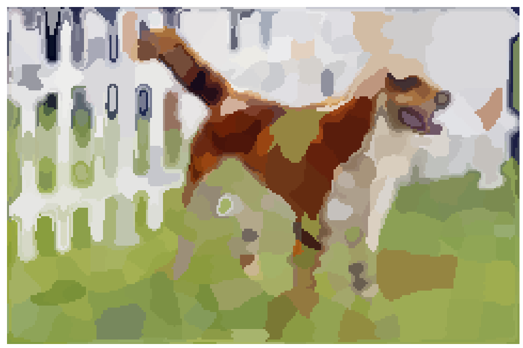

Figure 185: Image **tiger3** and the bounds of its segments. `(min_area, ncut_thresh, max_depth)` ≡ `(10, 0.5, 10). radius = 6`

Figure 186: Image **tiger3** segmented with `(min_area, ncut_thresh, max_depth)` ≡ `(10, 0.5, 10). radius = 3`

Figure 187: Image **tiger3** and the bounds of its segments. `(min_area, ncut_thresh, max_depth)` ≡ `(10, 0.5, 10). radius = 6`

Figure 188: Image `tiger3` segmented with `(min_area, ncut_thresh, max_depth)` $\equiv$ `(100, 0.1, 8). radius = 3`

Figure 189: Image `tiger3` and the bounds of its segments. `(min_area, ncut_thresh, max_depth)` $\equiv$ `(100, 0.1, 8). radius = 6`

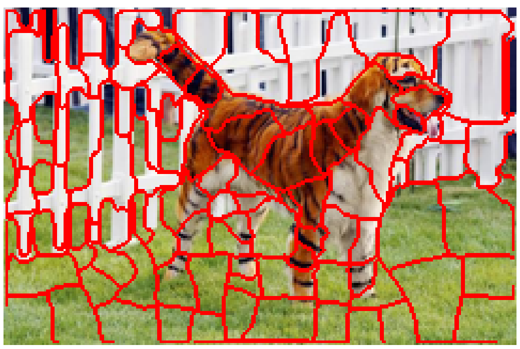

Figure 190: Image `tiger3` segmented with `(min_area, ncut_thresh, max_depth)` $\equiv$ `(100, 0.1, 8). radius = 3`

Figure 191: Image `tiger3` and the bounds of its segments. `(min_area, ncut_thresh, max_depth)` $\equiv$ `(100, 0.1, 8). radius = 6`

4 Segmentation using graph cuts

4.1 Preliminary results – `(`α`, `σ`)` $\equiv$ `(8.0, 10.0)`

4.1.1 Image `tiger1`

Figure 192: The image to be segmented into foreground and background.

Figure 193: Each pixel is assigned a probability proportional to the likelihood that it belongs to the foreground.

Figure 194: Image 193 thresholded to 0.5.

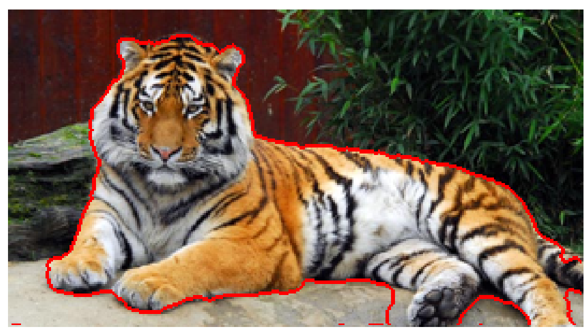

Figure 195: The resulting of the graph cut segmetation method on image `tiger1`.

4.1.2 Image **orange**

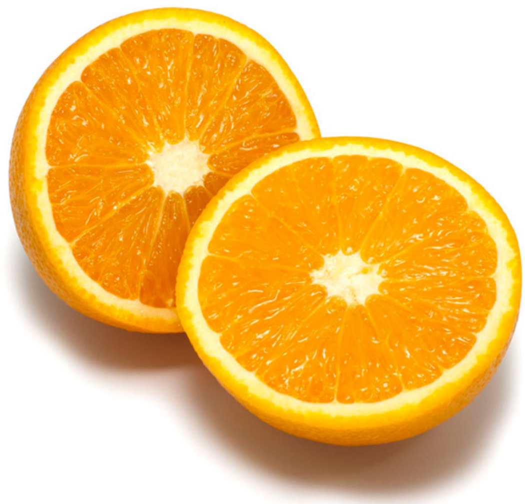

Figure 196: The image to be segmented into foreground and background.

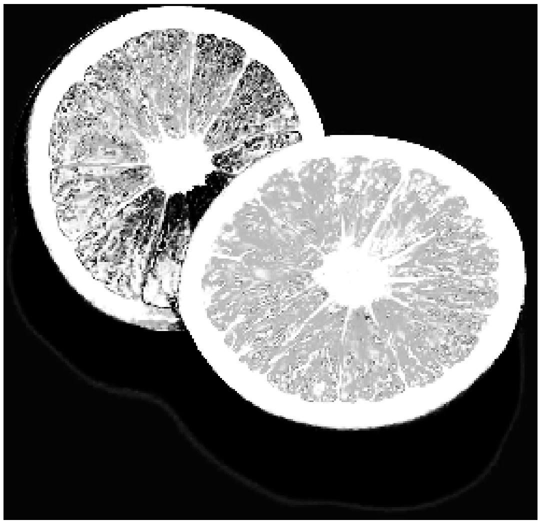

Figure 197: Each pixel is assigned a probability proportional to the likelihood that it belongs to the foreground.

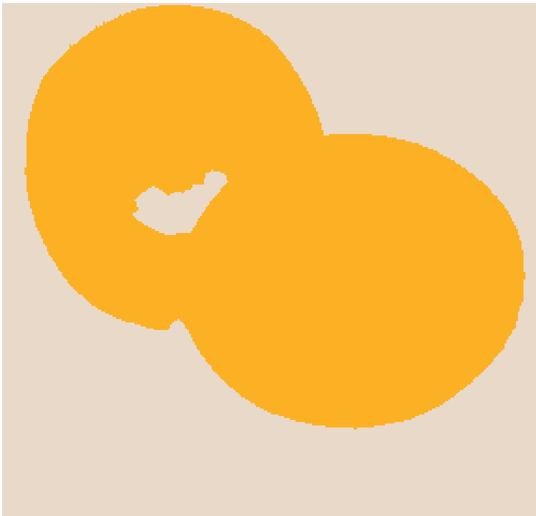

Figure 198: Image 197 thresholded to 0.5.

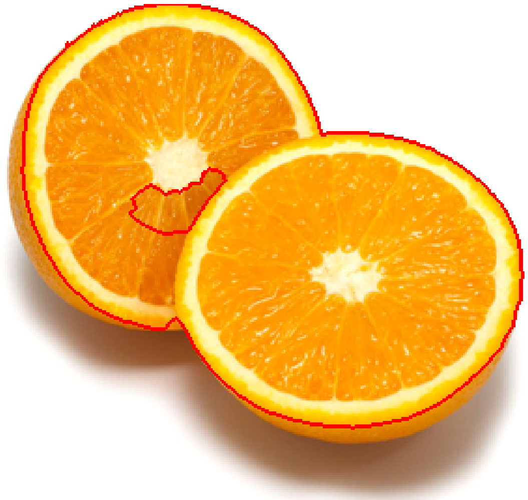

Figure 199: The resulting of the graph cut segmetation method on image `orange`.

4.1.3 Image **tiger2**

Figure 200: The image to be segmented into foreground and background.

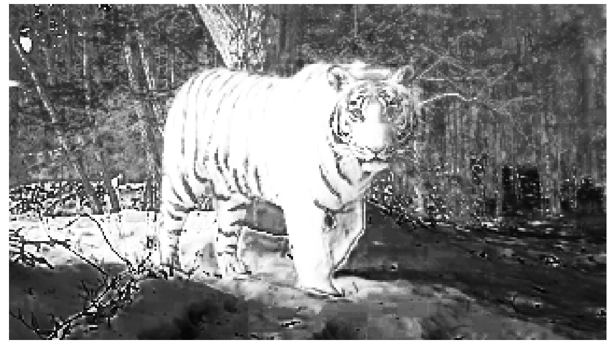

Figure 201: Each pixel is assigned a probability proportional to the likelihood that it belongs to the foreground.

Figure 202: Image 201 thresholded to 0.5.

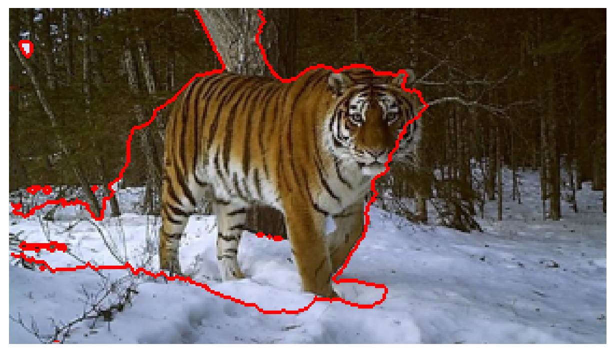

Figure 203: The resulting of the graph cut segmetation method on image `tiger2`.

4.1.4 Image **tiger3**

Figure 204: The image to be segmented into foreground and background.

Figure 205: Each pixel is assigned a probability proportional to the likelihood that it belongs to the foreground.

Figure 206: Image 205 thresholded to 0.5.

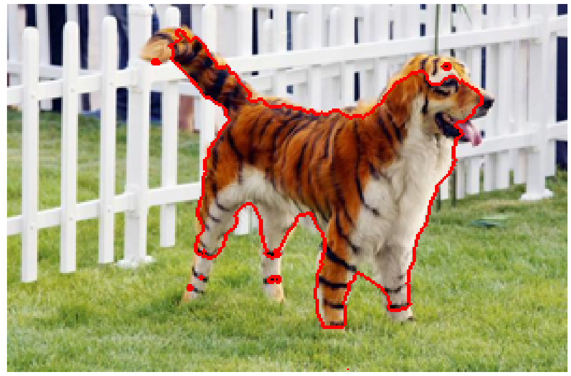

Figure 207: The resulting of the graph cut segmetation method on image `tiger3`.

4.2 Varying **(α, σ)**

4.2.1 Image **orange**

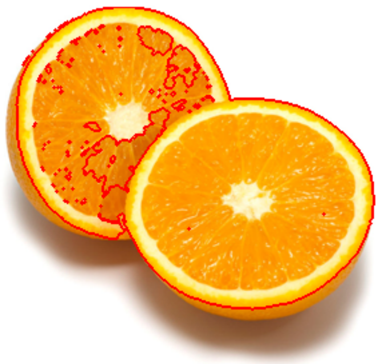

Figure 208: The resulting of the graph cut segmetation method on image `orange` for (α, σ)$\equiv$ (2,4)

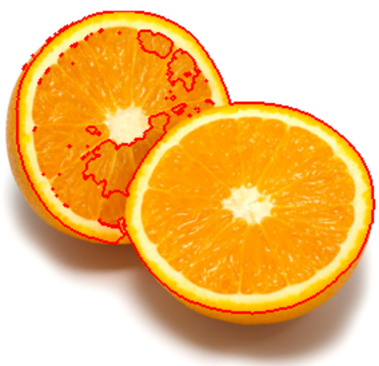

Figure 209: The resulting of the graph cut segmetation method on image `orange` for (α, σ)$\equiv$ (2,6)

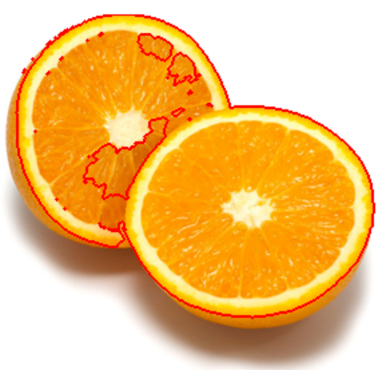

Figure 210: The resulting of the graph cut segmetation method on image `orange` for (α, σ)$\equiv$ (2,10)

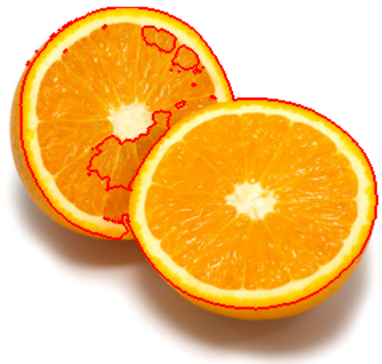

Figure 211: The resulting of the graph cut segmetation method on image `orange` for (α, σ)$\equiv$ (2,12)

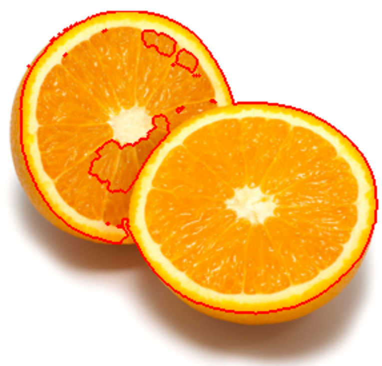

Figure 212: The resulting of the graph cut segmetation method on image `orange` for (α, σ)$\equiv$ (2,16)

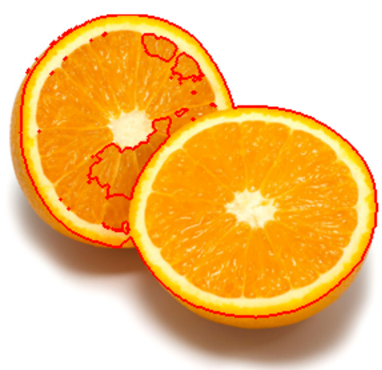

Figure 213: The resulting of the graph cut segmetation method on image `orange` for (α, σ)$\equiv$ (4,4)

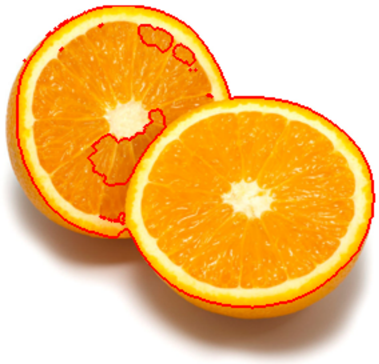

Figure 214: The resulting of the graph cut segmetation method on image `orange` for $(\alpha,\ \sigma)\equiv$ (4,6)

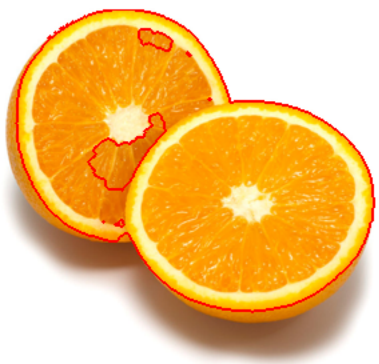

Figure 215: The resulting of the graph cut segmetation method on image `orange` for $(\alpha,\ \sigma)\equiv$ (4,10)

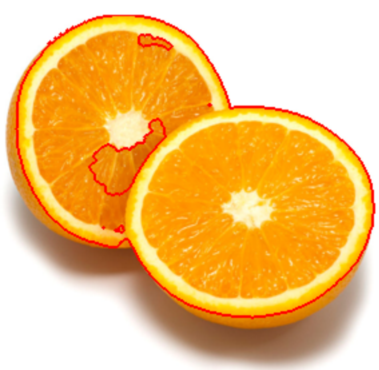

Figure 216: The resulting of the graph cut segmetation method on image `orange` for $(\alpha,\ \sigma)\equiv$ (4,12)

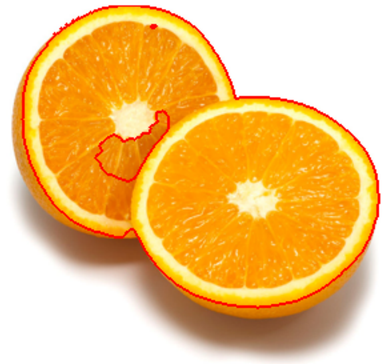

Figure 217: The resulting of the graph cut segmetation method on image `orange` for $(\alpha,\ \sigma)\equiv$ (4,16)

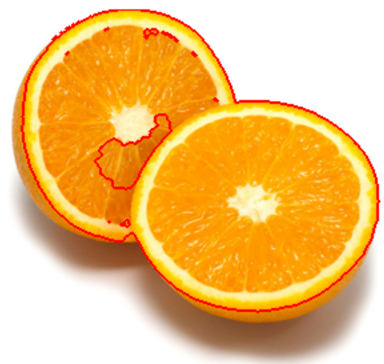

Figure 218: The resulting of the graph cut segmetation method on image `orange` for $(\alpha,\ \sigma)\equiv$ (8,4)

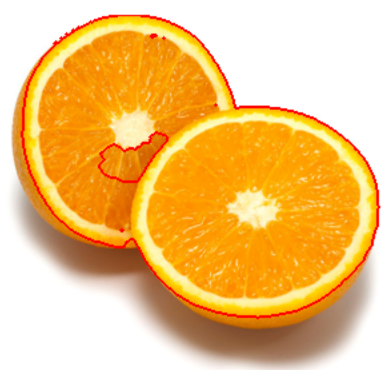

Figure 219: The resulting of the graph cut segmetation method on image `orange` for $(\alpha,\ \sigma)\equiv$ (8,6)

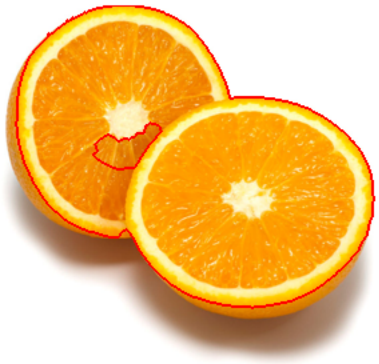

Figure 220: The resulting of the graph cut segmetation method on image `orange` for $(\alpha,\ \sigma)\equiv$ (8,10)

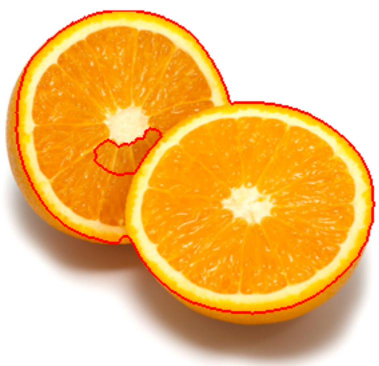

Figure 221: The resulting of the graph cut segmetation method on image `orange` for $(\alpha,\ \sigma)\equiv$ (8,12)

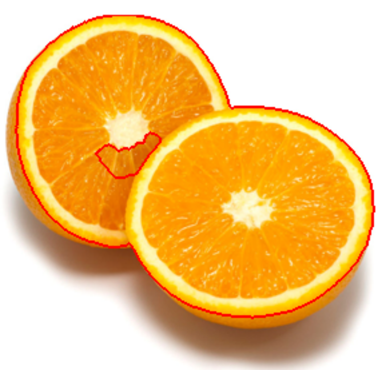

Figure 222: The resulting of the graph cut segmetation method on image `orange` for $(\alpha,\ \sigma)\equiv$ (8,16)

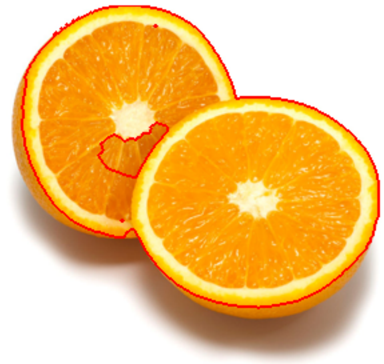

Figure 223: The resulting of the graph cut segmetation method on image `orange` for $(\alpha,\ \sigma)\equiv$ (12,4)

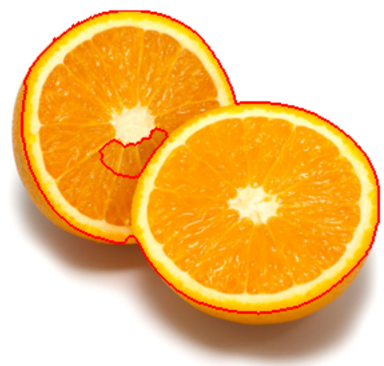

Figure 224: The resulting of the graph cut segmetation method on image `orange` for $(\alpha,\ \sigma)\equiv$ (12,6)

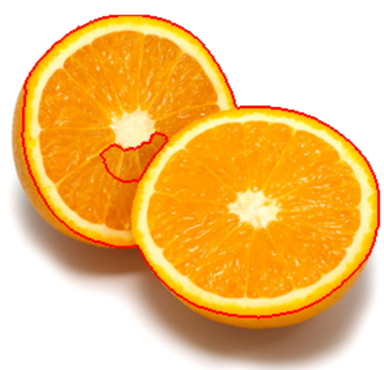

Figure 225: The resulting of the graph cut segmetation method on image `orange` for $(\alpha,\ \sigma)\equiv$ (12,10)

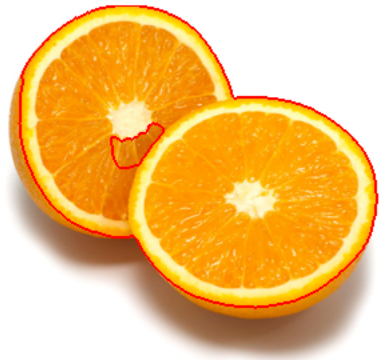

Figure 226: The resulting of the graph cut segmetation method on image `orange` for (α, σ)$\equiv$ (12,12)

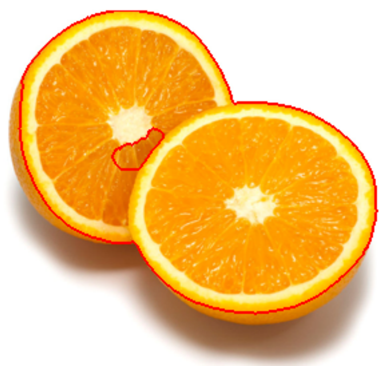

Figure 227: The resulting of the graph cut segmetation method on image `orange` for (α, σ)$\equiv$ (12,16)

4.2.2 Image `tiger1`

Figure 228: The resulting of the graph cut segmetation method on image `tiger1` for (α, σ)$\equiv$ `(2,4)`

Figure 229: The resulting of the graph cut segmetation method on image `tiger1` for (α, σ)$\equiv$ `(2,6)`

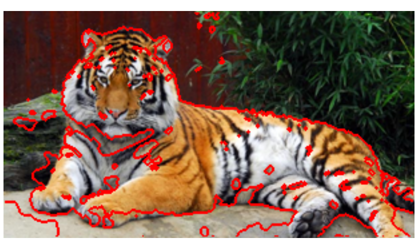

Figure 230: The resulting of the graph cut segmetation method on image `tiger1` for (α, σ)$\equiv$ `(2,10)`

Figure 231: The resulting of the graph cut segmetation method on image `tiger1` for (α, σ)$\equiv$ `(2,12)`

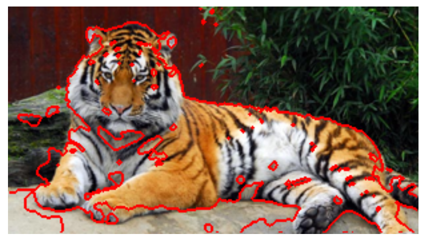

Figure 232: The resulting of the graph cut segmetation method on image `tiger1` for (α, σ)$\equiv$ `(2,16)`

Figure 233: The resulting of the graph cut segmetation method on image `tiger1` for (α, σ)$\equiv$ `(4,4)`

Figure 234: The resulting of the graph cut segmetation method on image `tiger1` for $(\alpha, \sigma) \equiv$ (4,6)

Figure 235: The resulting of the graph cut segmetation method on image `tiger1` for $(\alpha, \sigma) \equiv$ (4,10)

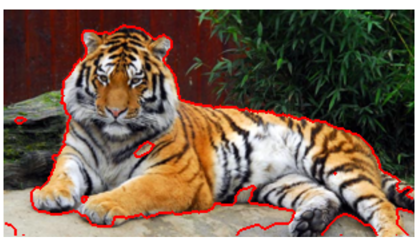

Figure 236: The resulting of the graph cut segmetation method on image `tiger1` for $(\alpha, \sigma) \equiv$ (4,12)

Figure 237: The resulting of the graph cut segmetation method on image `tiger1` for $(\alpha, \sigma) \equiv$ (4,16)

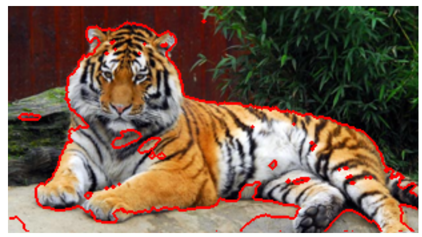

Figure 238: The resulting of the graph cut segmetation method on image `tiger1` for $(\alpha, \sigma) \equiv$ (8,4)

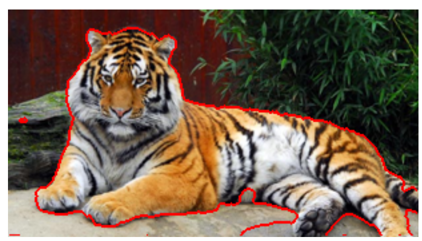

Figure 239: The resulting of the graph cut segmetation method on image `tiger1` for $(\alpha, \sigma) \equiv$ (8,6)

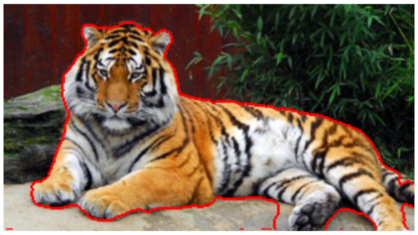

Figure 240: The resulting of the graph cut segmetation method on image `tiger1` for (α, σ)$\equiv$ (8,10)

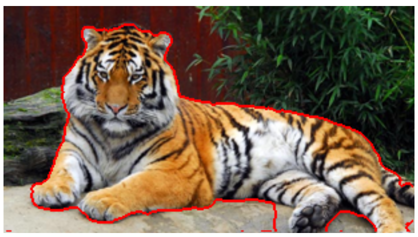

Figure 241: The resulting of the graph cut segmetation method on image `tiger1` for (α, σ)$\equiv$ (8,12)

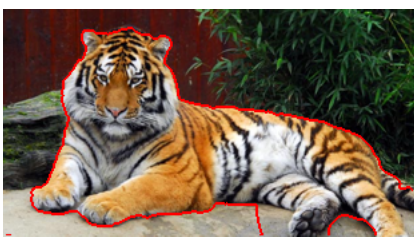

Figure 242: The resulting of the graph cut segmetation method on image `tiger1` for (α, σ)$\equiv$ (8,16)

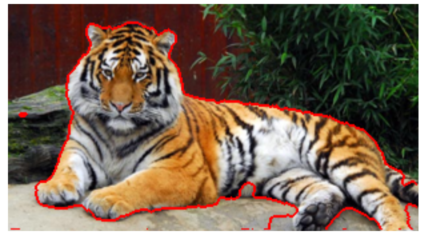

Figure 243: The resulting of the graph cut segmetation method on image `tiger1` for (α, σ)$\equiv$ (12,4)

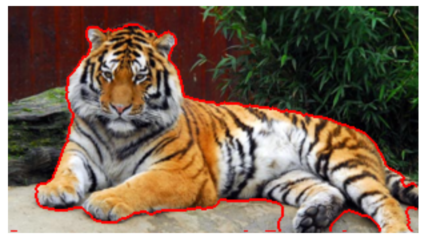

Figure 244: The resulting of the graph cut segmetation method on image `tiger1` for (α, σ)$\equiv$ (12,6)

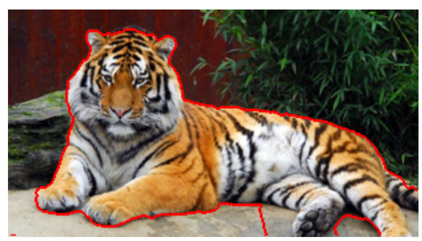

Figure 245: The resulting of the graph cut segmetation method on image `tiger1` for (α, σ)$\equiv$ (12,10)

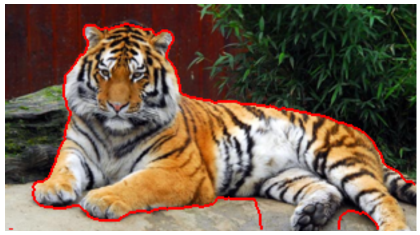

Figure 246: The resulting of the graph cut segmetation method on image `tiger1` for `(`α`, `σ`)`$\equiv$` (12,12)`

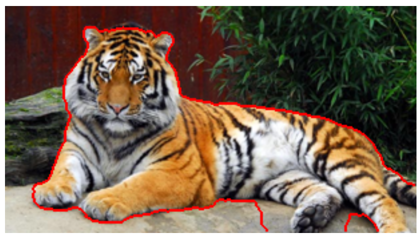

Figure 247: The resulting of the graph cut segmetation method on image `tiger1` for `(`α`, `σ`)`$\equiv$` (12,16)`

4.2.3 Image **tiger2**

Figure 248: The resulting of the graph cut segmetation method on image `tiger2` for `(`α`,` σ`)`$\equiv$ `(2,4)`

Figure 249: The resulting of the graph cut segmetation method on image `tiger2` for `(`α`,` σ`)`$\equiv$ `(2,6)`

Figure 250: The resulting of the graph cut segmetation method on image `tiger2` for `(`α`,` σ`)`$\equiv$ `(2,10)`

Figure 251: The resulting of the graph cut segmetation method on image `tiger2` for `(`α`,` σ`)`$\equiv$ `(2,12)`

Figure 252: The resulting of the graph cut segmetation method on image `tiger2` for `(`α`,` σ`)`$\equiv$ `(2,16)`

Figure 253: The resulting of the graph cut segmetation method on image `tiger2` for `(`α`,` σ`)`$\equiv$ `(4,4)`

Figure 254: The resulting of the graph cut segmetation method on image `tiger2` for (α, σ)$\equiv$ (4,6)

Figure 255: The resulting of the graph cut segmetation method on image `tiger2` for (α, σ)$\equiv$ (4,10)

Figure 256: The resulting of the graph cut segmetation method on image `tiger2` for (α, σ)$\equiv$ (4,12)

Figure 257: The resulting of the graph cut segmetation method on image `tiger2` for (α, σ)$\equiv$ (4,16)

Figure 258: The resulting of the graph cut segmetation method on image `tiger2` for (α, σ)$\equiv$ (8,4)

Figure 259: The resulting of the graph cut segmetation method on image `tiger2` for (α, σ)$\equiv$ (8,6)

Figure 260: The resulting of the graph cut segmetation method on image `tiger2` for (α, σ)$\equiv$ (8,10)

Figure 261: The resulting of the graph cut segmetation method on image `tiger2` for (α, σ)$\equiv$ (8,12)

Figure 262: The resulting of the graph cut segmetation method on image `tiger2` for (α, σ)$\equiv$ (8,16)

Figure 263: The resulting of the graph cut segmetation method on image `tiger2` for (α, σ)$\equiv$ (12,4)

Figure 264: The resulting of the graph cut segmetation method on image `tiger2` for (α, σ)$\equiv$ (12,6)

Figure 265: The resulting of the graph cut segmetation method on image `tiger2` for (α, σ)$\equiv$ (12,10)

Figure 266: The resulting of the graph cut segmetation method on image `tiger2` for $(\alpha, \sigma) \equiv$ `(12,12)`

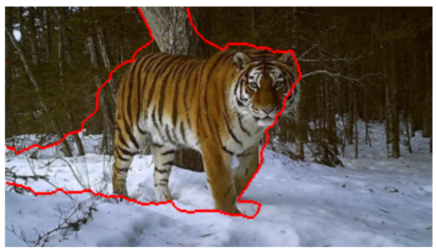

Figure 267: The resulting of the graph cut segmetation method on image `tiger2` for $(\alpha, \sigma) \equiv$ `(12,16)`

4.2.4 Image **tiger3**

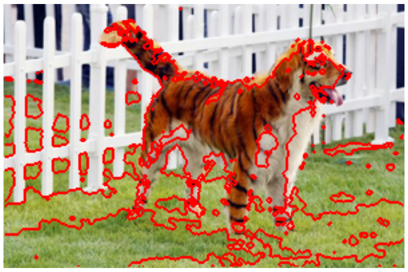

Figure 268: The resulting of the graph cut segmetation method on image `tiger3` for $(\alpha,\ \sigma)\equiv$ (2,4)

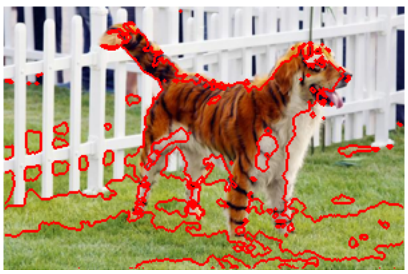

Figure 269: The resulting of the graph cut segmetation method on image `tiger3` for $(\alpha,\ \sigma)\equiv$ (2,6)

Figure 270: The resulting of the graph cut segmetation method on image `tiger3` for $(\alpha,\ \sigma)\equiv$ (2,10)

Figure 271: The resulting of the graph cut segmetation method on image `tiger3` for $(\alpha,\ \sigma)\equiv$ (2,12)

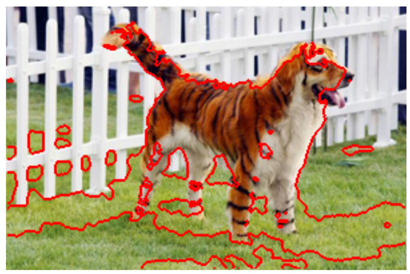

Figure 272: The resulting of the graph cut segmetation method on image `tiger3` for $(\alpha,\ \sigma)\equiv$ (2,16)

Figure 273: The resulting of the graph cut segmetation method on image `tiger3` for $(\alpha,\ \sigma)\equiv$ (4,4)

Figure 274: The resulting of the graph cut segmetation method on image `tiger3` for $(\alpha,\ \sigma)\equiv$ (4,6)

Figure 275: The resulting of the graph cut segmetation method on image `tiger3` for $(\alpha,\ \sigma)\equiv$ (4,10)

Figure 276: The resulting of the graph cut segmetation method on image `tiger3` for $(\alpha,\ \sigma)\equiv$ (4,12)

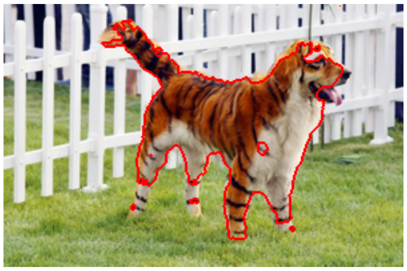

Figure 277: The resulting of the graph cut segmetation method on image `tiger3` for $(\alpha,\ \sigma)\equiv$ (4,16)

Figure 278: The resulting of the graph cut segmetation method on image `tiger3` for $(\alpha,\ \sigma)\equiv$ (8,4)

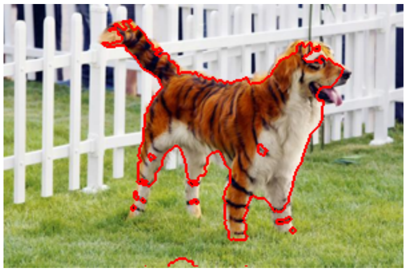

Figure 279: The resulting of the graph cut segmetation method on image `tiger3` for $(\alpha,\ \sigma)\equiv$ (8,6)

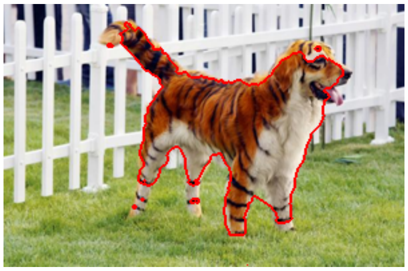

Figure 280: The resulting of the graph cut segmetation method on image `tiger3` for $(\alpha,\ \sigma)\equiv$ (8,10)

Figure 283: The resulting of the graph cut segmetation method on image `tiger3` for $(\alpha,\ \sigma)\equiv$ (12,4)

Figure 281: The resulting of the graph cut segmetation method on image `tiger3` for $(\alpha,\ \sigma)\equiv$ (8,12)

Figure 284: The resulting of the graph cut segmetation method on image `tiger3` for $(\alpha,\ \sigma)\equiv$ (12,6)

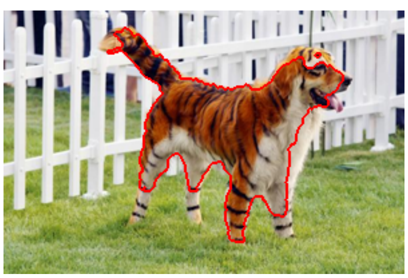

Figure 282: The resulting of the graph cut segmetation method on image `tiger3` for $(\alpha,\ \sigma)\equiv$ (8,16)

Figure 285: The resulting of the graph cut segmetation method on image `tiger3` for $(\alpha,\ \sigma)\equiv$ (12,10)

Figure 286: The resulting of the graph cut segmetation method on image `tiger3` for (α, σ)$\equiv$ (12,12)

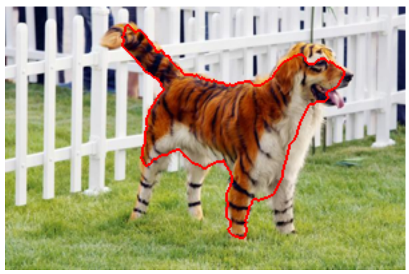

Figure 287: The resulting of the graph cut segmetation method on image `tiger3` for (α, σ)$\equiv$ (12,16)

4.3 Question 11

Images 208 - 287 illustrate the resulting segmentation of images `orange, tiger{1,2,3}` for all combinations of $\alpha = [2.0, 4.0, 8.0, 12.0]$ and $\sigma = [4.0, 6.0, 10.0, 12.0, 16.0]$.

From these images we can deduce that the optimal setting for variables α, σ is in the vicinity of $(8.0, 10.0)$. For a higher values the resulting segmentation remains almost the same, although it becomes coarse. The opposite can be said for smaller values of α, σ: the lower their values, the more sensitive and less accurate becomes the division between the foreground and background pixels.

4.4 Question 12

Figures 288 - 303 illustrate the resulting segmentation of image `tiger1` for values of `K` $\in$ `[1, 16], K` $\in \mathbb{Z}$.

As is evident from these images, the minimum value for `K` that can still produce reasonably accurate segmentation results for image `tiger1` is `K=6`.

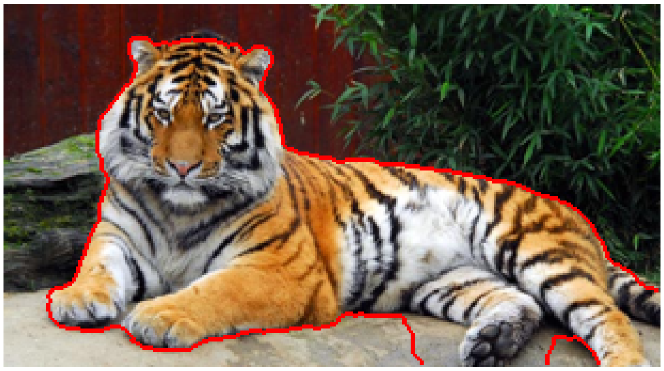

Figure 288: Image `tiger1` segmented with `K = 16`

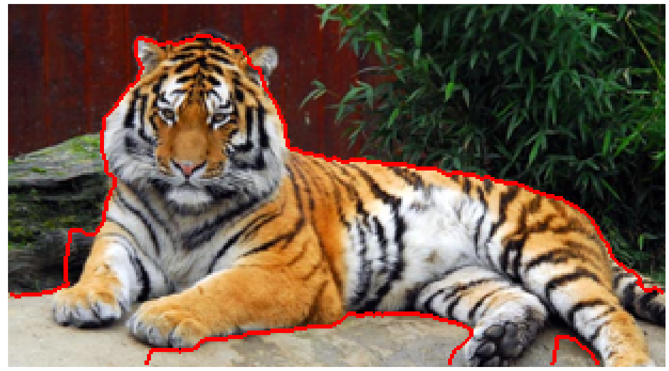

Figure 289: Image `tiger1` segmented with `K = 15`

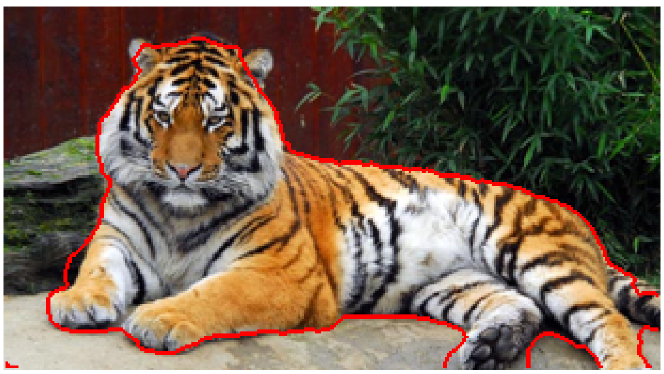

Figure 290: Image `tiger1` segmented with `K = 14`

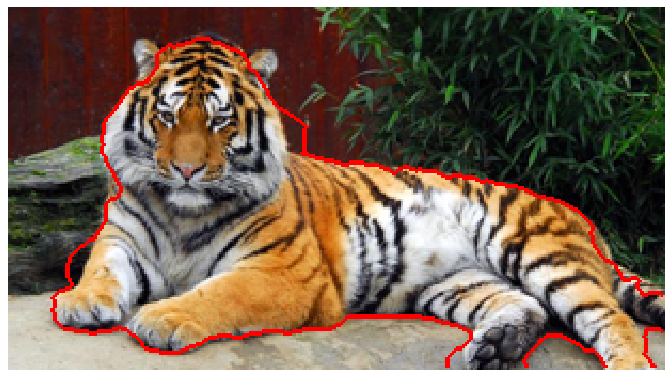

Figure 291: Image `tiger1` segmented with `K = 13`

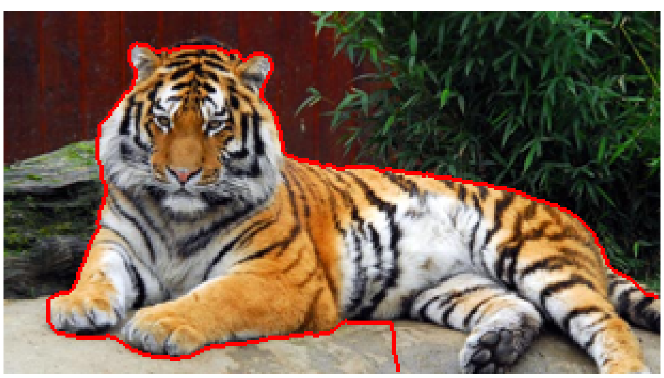

Figure 292: Image `tiger1` segmented with `K = 12`

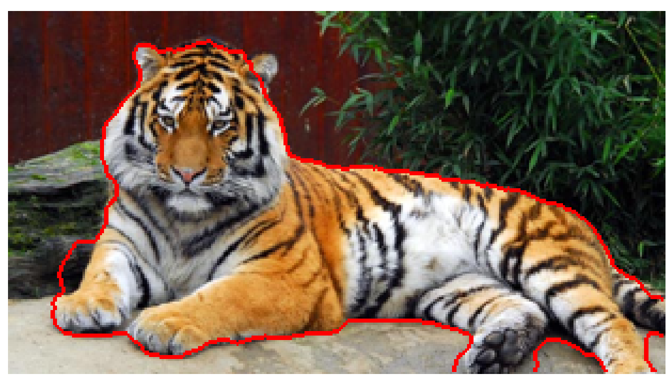

Figure 293: Image `tiger1` segmented with `K = 11`

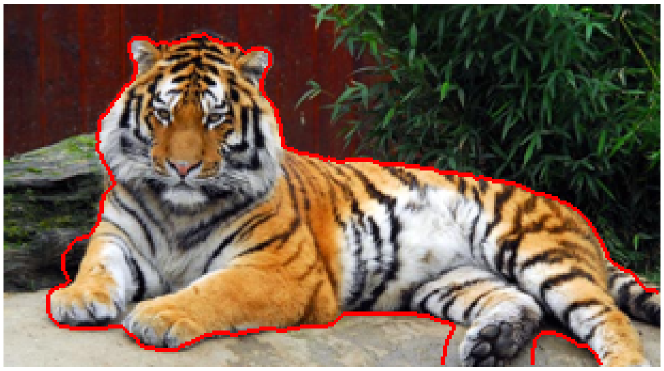

Figure 294: Image `tiger1` segmented with `K = 10`

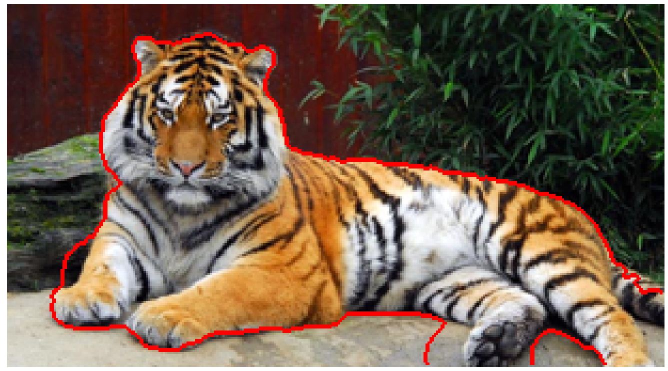

Figure 295: Image `tiger1` segmented with `K = 9`

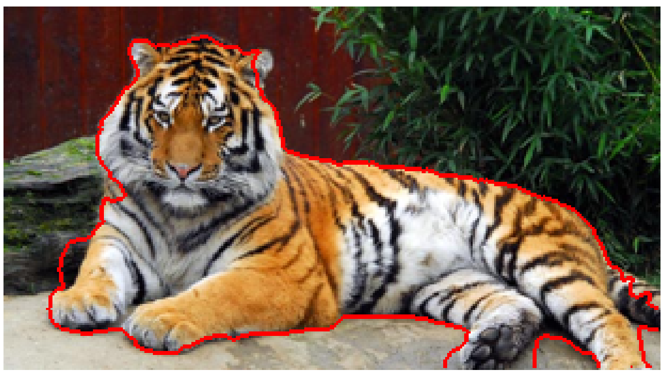

Figure 296: Image `tiger1` segmented with `K = 8`

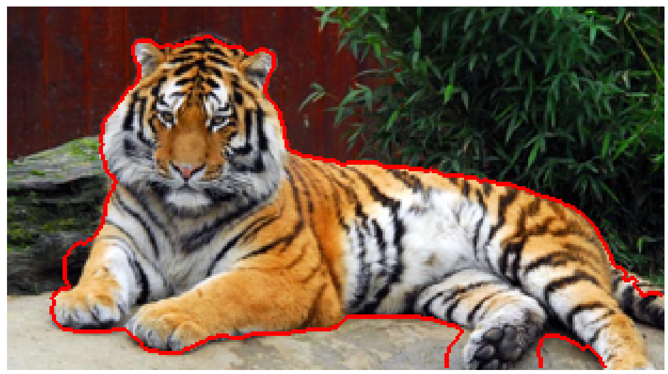

Figure 297: Image `tiger1` segmented with `K = 7`

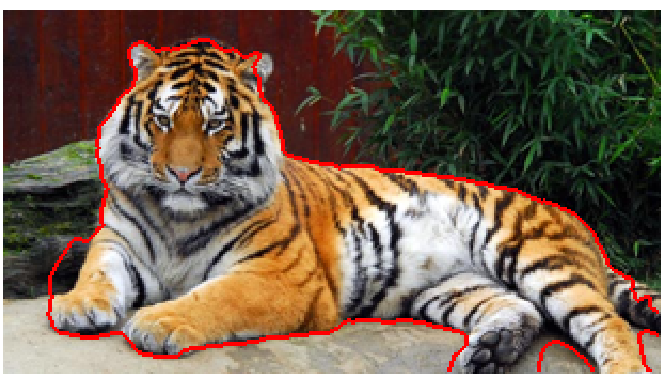

Figure 298: Image `tiger1` segmented with `K = 6`

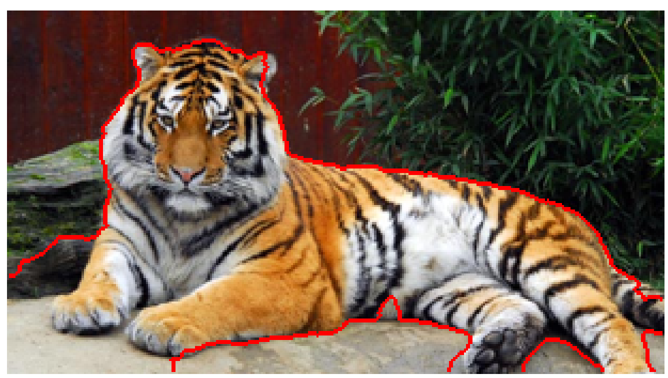

Figure 299: Image `tiger1` segmented with `K = 5`

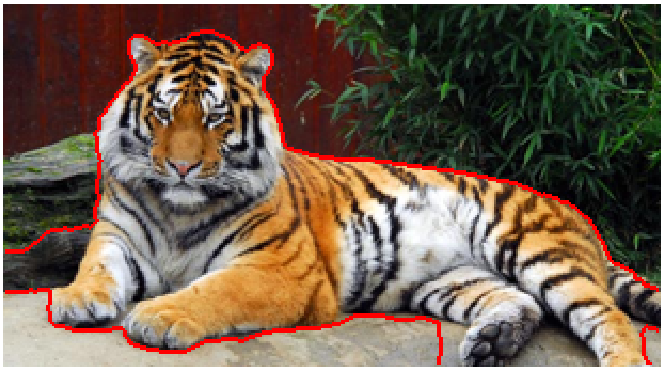

Figure 300: Image `tiger1` segmented with `K = 4`

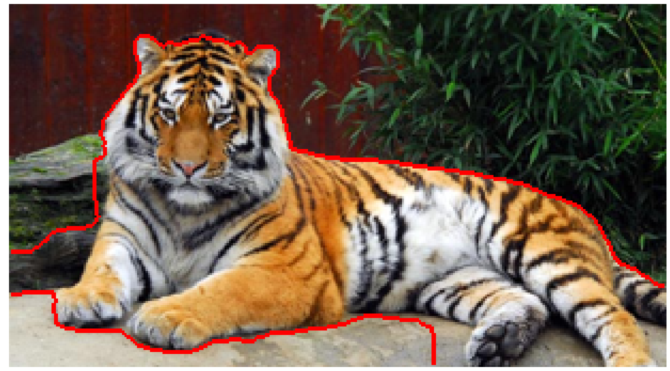

Figure 301: Image `tiger1` segmented with `K = 3`

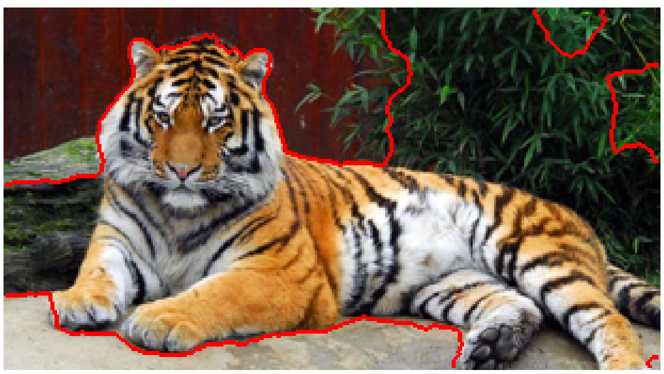

Figure 302: Image `tiger1` segmented with `K = 2`

Figure 303: Image `tiger1` segmented with `K = 1`

4.5 Question 13

It depends on the context of the application that utilizes this method. For instance, in image classification it would be helpful to provide a bounding box on some objects so as to obtain an accurate training set of the classes that the instances of the test dataset will be sorted into.

However, if the application at hand uses massive amounts of heterogenous settings or no clear identification objects, it would be unwise to use this method.

Furthermore, if the earlier segmentation methods and Graph Cuts is used for the same purpose, it is a matter of the desired accuracy which segmentation method is to be preferred. The latter provides a coarser (in terms of colour representation) but more accurate (in terms of colour similarity) segmentation.

4.6 Question 14

Starting from the outside in, all considered segmentation methods try to find similarities in colour and group pixels deemed similar into one segment. That said, however, from the 4 considered segmentation methods only `K-means` has no spatial awareness, and thus its output segments typically span across split-up areas. Although `mean-shift` does take spatial provisions, its modus operandi is closer to `K-means` than `Normalized Cut` or `Graph cut` who treat an image as a graph and build their approach from there. Their main difference, and indeed the main difference between `Graph cut` and all the other segmentation methods is that it employs prior information in order to deliver more accurate segments.