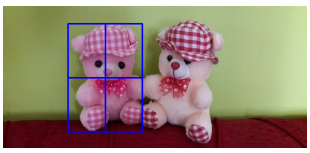
Grab Cut: A study of how changes in parameters affects Segmentation

Swetanjal Murati Dutta
International Institute of Information Technology, Hyderabad
Gachibowli, Telangana
swetanjal.dutta@research.iiit.ac.in



(a) Original Image

(b) Segmented Image

Figure 1: Illustration of image segmentation results on my own images

Abstract

This report presents an analysis of how changes in various parameters affects image segmentation by the Grab Cut Algorithm.

1. Number of iterations of Gaussian Mixture Model updating and Energy Minimization

It is observed that as we increase the number of iterations we get more and more perfect segmentation of the foreground object until it converges. This is shown in Fig 1, Fig 2, Fig 3 and Fig 4. Note that in this section we keep all other parameters unchanged.

2. The number of Gaussian Mixture Components

Increasing the number of Gaussian Mixture components helps us in learning and representing a wider range of colors for the foreground and background. If the number of components is very low, it may result in incorrectly categorizing many foreground pixels if the foreground object shows wide variation in color combination. At the same time if there are too many components there can be additional background pixels present in the segmentation of foreground object. This is a hyper parameter that has to be tuned experimentally and there is no perfect rule for deciding the parameter.

3. Color Space

There is no effect in image segmentation by changing the color space.

4. Varying Gamma

Varying gamma is responsible for giving the pairwise potential values between any two pixels. It is a hyper parameter which has to be determined experimentally. Too low a value or too high a value can wrongly categorise the foreground pixels. Typically a value of 50 or 100 gave us decent results.

5. A Tight or Loose initial bounding box

A tighter initial bounding box results in a more stricter supervision and hence a more accurate segmentation of the foreground object from the background.

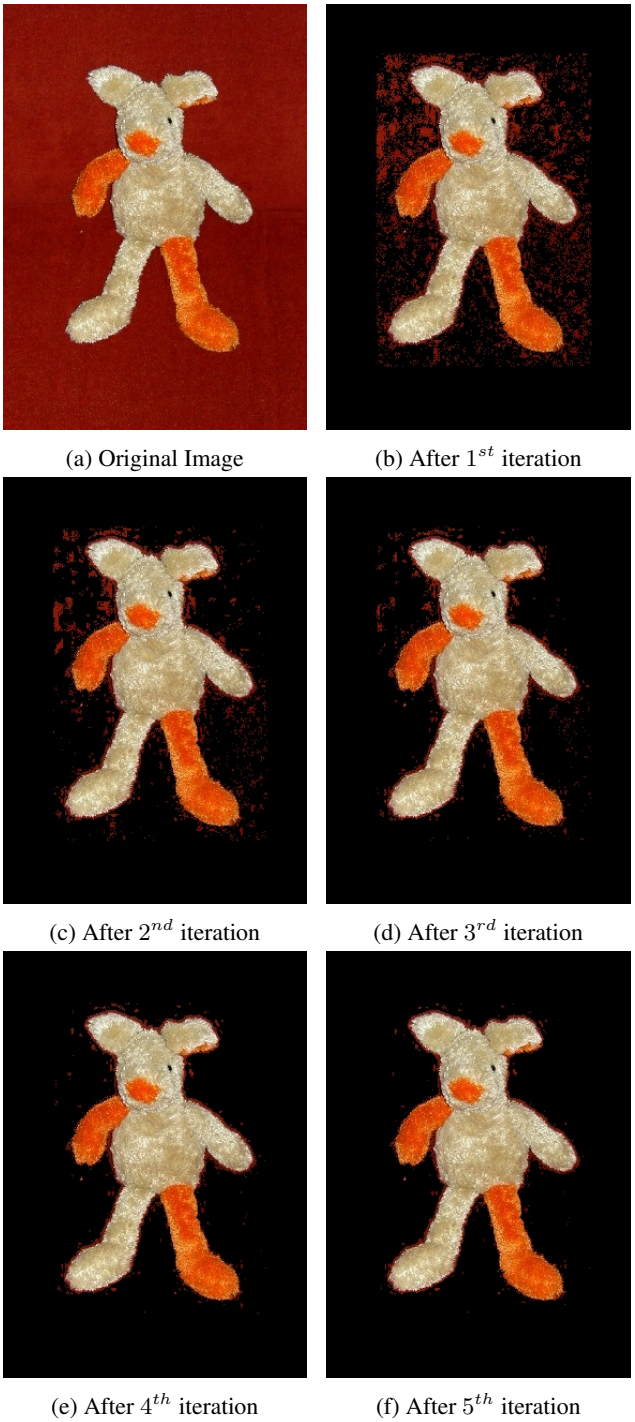
A loose initial bounding box will result in a weaker supervision and hence a lot of background pixels would be present in the region of interest which wouldn't be learned as a part of background model and hence we won't get a very accurate segmentation.

6. Better Ways of using the bounding box for segmentation

We can use a brush like structure as present in MS Paint to provide a stricter supervision at the pixel level. This is because a rectangular bounding box may not provide accurate enough supervision depending on the shape of the object. Apart from this, we can have bounding boxes of different shapes and also give the option of choosing multiple region of interests.

7. 4 Neighbourhood vs 8 Neighbourhood

Using a 8 neighbourhood would result in a more continuous segmentation as the diagonal pixels would also influence the segmentation.

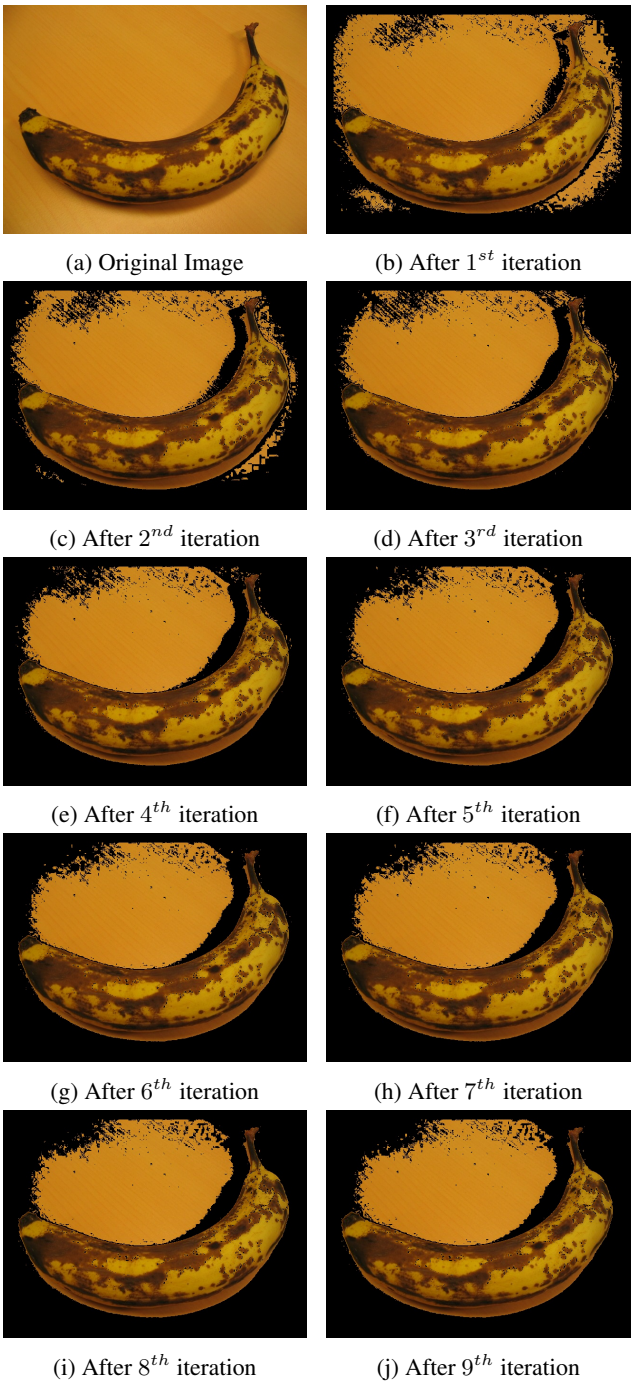


(a) Original Image (b) After 1^{st} iteration

(c) After 2^{nd} iteration (d) After 3^{rd} iteration

(e) After 4^{th} iteration (f) After 5^{th} iteration

Figure 2: Illustration of image segmentation results for various number of iterations.



(a) Original Image (b) After 1^{st} iteration

(c) After 2^{nd} iteration (d) After 3^{rd} iteration

(e) After 4^{th} iteration (f) After 5^{th} iteration

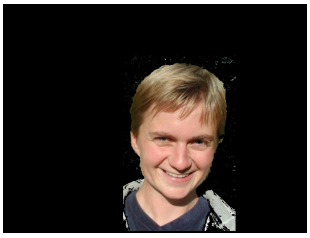
(g) After 6^{th} iteration (h) After 7^{th} iteration

(i) After 8^{th} iteration (j) After 9^{th} iteration

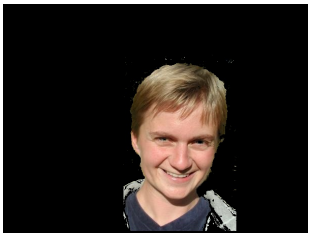
Figure 3: Illustration of image segmentation results for various number of iterations.

(a) Original Image

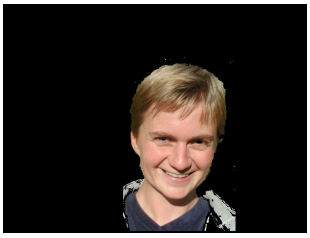
(b) After 1^{st} iteration



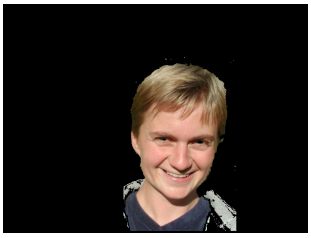
(c) After 2^{nd} iteration



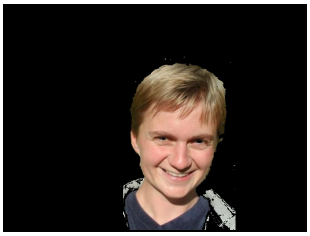
(d) After 3^{rd} iteration



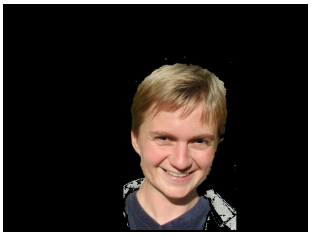
(e) After 4^{th} iteration



(f) After 5^{th} iteration



(g) After 6^{th} iteration



(h) After 7^{th} iteration

Figure 4: Illustration of image segmentation results for various number of iterations.

(a) Original Image

(b) After 1^{st} iteration

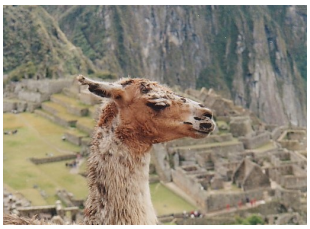
(c) After 2^{nd} iteration

(d) After 3^{rd} iteration

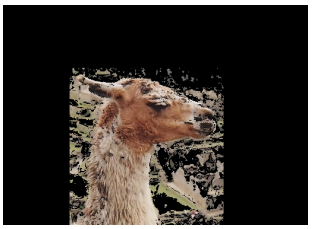
(e) After 4^{th} iteration

(f) After 5^{th} iteration

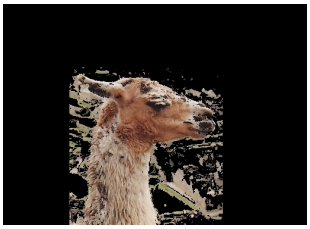
Figure 5: Illustration of image segmentation results for various number of iterations.



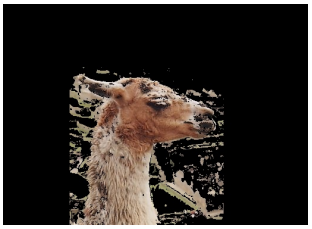
(a) Original Image



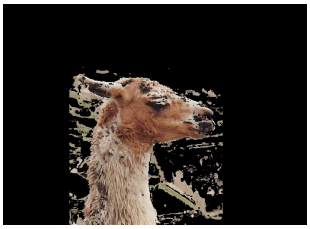
(b) After 1^{st} iteration



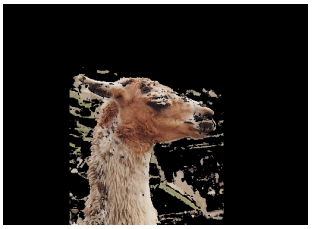
(c) After 2^{nd} iteration



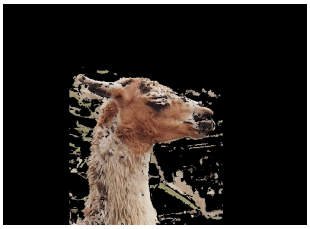
(d) After 3^{rd} iteration



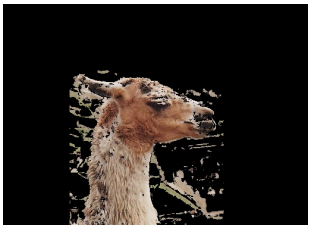
(e) After 4^{th} iteration



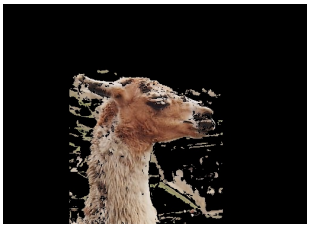
(f) After 5^{th} iteration



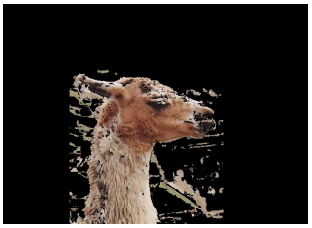
(g) After 6^{th} iteration



(h) After 7^{th} iteration



(i) After 8^{th} iteration



(j) After 9^{th} iteration

Figure 6: Illustration of image segmentation results for various number of iterations.

(a) Original Image

(b) After 1^{st} iteration

(c) After 2^{nd} iteration

(d) After 3^{rd} iteration

(e) After 4^{th} iteration

(f) After 5^{th} iteration

(g) After 6^{th} iteration

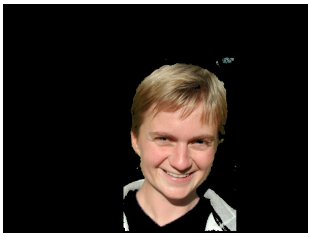
(h) After 7^{th} iteration

(i) After 9^{th} iteration

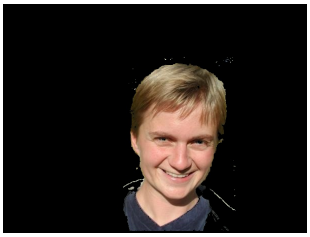
(j) After 10^{th} iteration

Figure 7: Illustration of image segmentation results for various number of iterations.

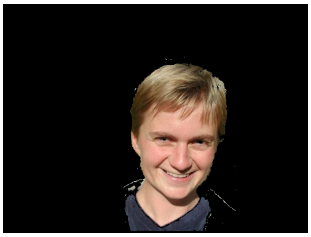
(a) Original Image



(b) 1 GMM Component

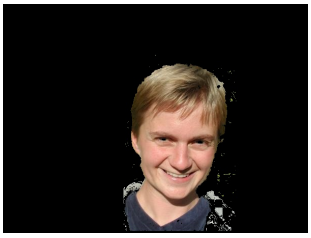


(c) 3 GMM Components



(d) 5 GMM Components

(e) 11 GMM Components

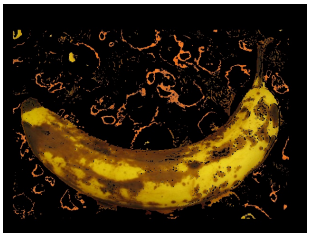


(f) 31 GMM Components

Figure 8: Illustration of image segmentation results for various number of GMM Components.

(a) Original Image

(b) 1 GMM Component

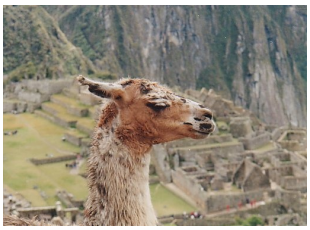


(c) 3 GMM Components

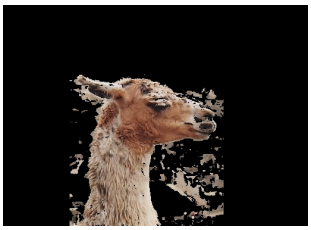
(d) 5 GMM Components

(e) 11 GMM Components

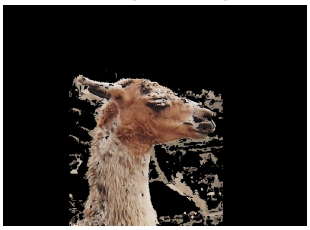
Figure 9: Illustration of image segmentation results for various number of GMM Components.



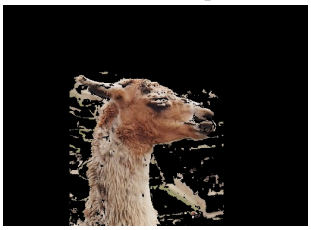
(a) Original Image



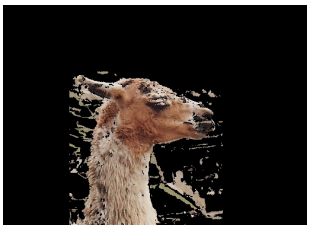
(b) 1 GMM Component



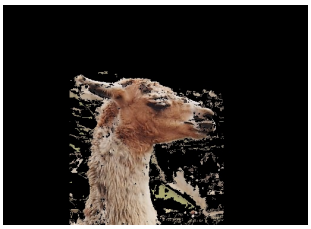
(c) 3 GMM Components



(d) 5 GMM Components



(e) 11 GMM Components



(f) 31 GMM Components

Figure 10: Illustration of image segmentation results for various number of GMM Components.

(a) Original Image

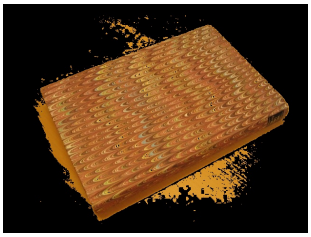
(b) RGB Color Space

(c) LAB Color Space

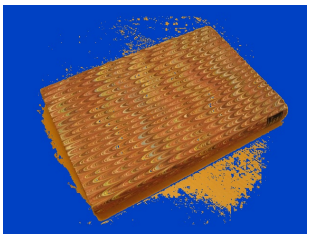
(d) YCrCb Color Space

Figure 11: Illustration of image segmentation results for various color spaces.

(a) Original Image



(b) RGB Color Space

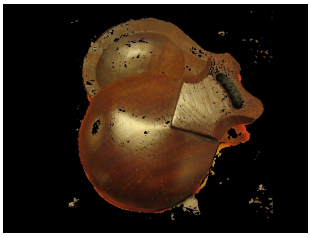


(c) LAB Color Space

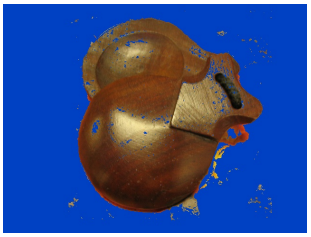
(d) YCrCb Color Space

Figure 12: Illustration of image segmentation results for various color spaces.

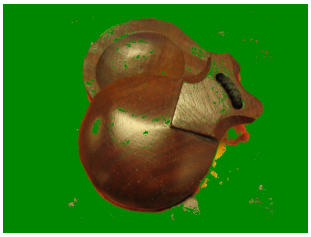
(a) Original Image



(b) RGB Color Space

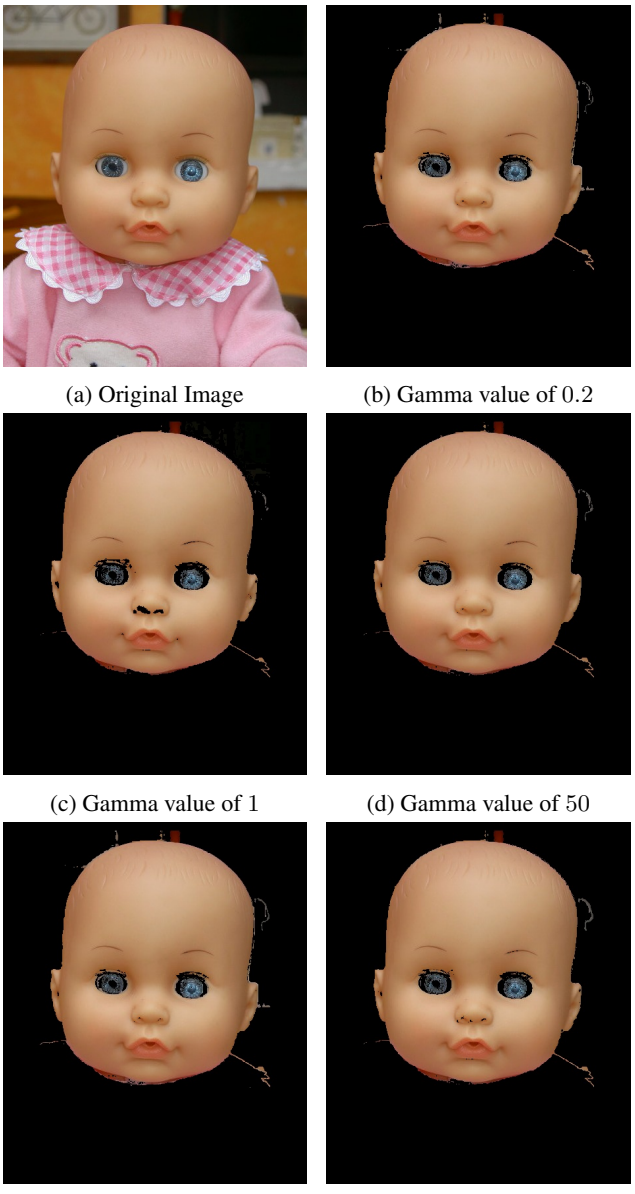


(c) LAB Color Space



(d) YCrCb Color Space

Figure 13: Illustration of image segmentation results for various color spaces.

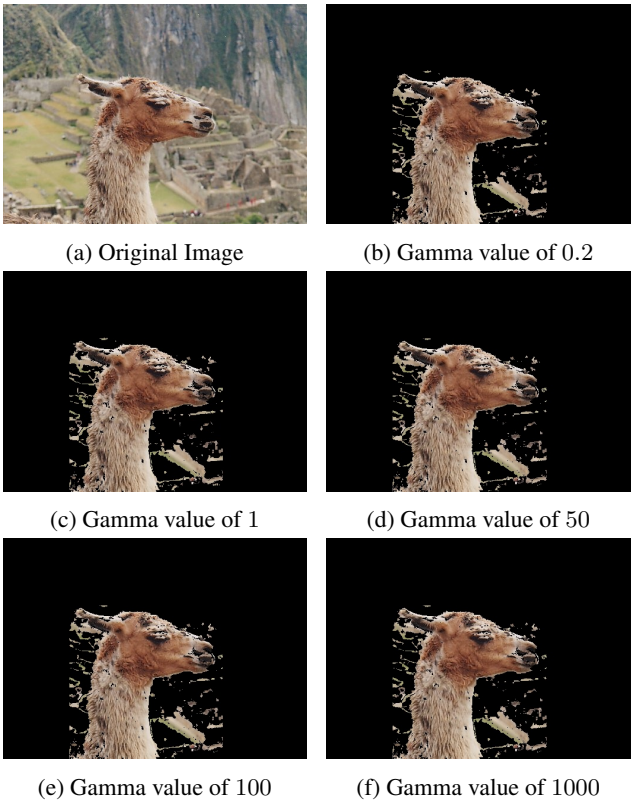


(a) Original Image (b) Gamma value of 0.2

(c) Gamma value of 1 (d) Gamma value of 50

(e) Gamma value of 100 (f) Gamma value of 1000

Figure 14: Illustration of image segmentation results for various values of gamma.



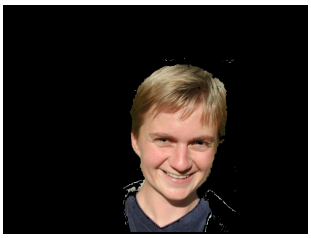
(a) Original Image (b) Gamma value of 0.2

(c) Gamma value of 1 (d) Gamma value of 50

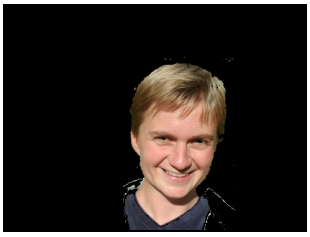
(e) Gamma value of 100 (f) Gamma value of 1000

Figure 15: Illustration of image segmentation results for various values of gamma.

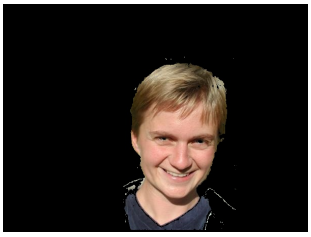
(a) Original Image



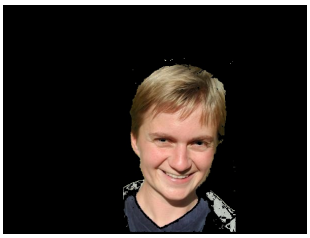
(b) Gamma value of 0.2



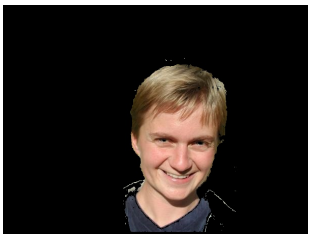
(c) Gamma value of 1



(d) Gamma value of 50



(e) Gamma value of 100



(f) Gamma value of 1000

Figure 16: Illustration of image segmentation results for various values of gamma.

(a) Original Image

(b) 4 Neighbourhood

(c) 8 Neighbourhood

Figure 17: Illustration of image segmentation results for 8 neighbourhood and 4 neighbourhood.