1.1 Gradient Descent

1.1.a No Regularization Gradient Descent

When using the Gradient Descent Algorithm with **no** *Ragularization Term* ($\lambda = 0$) we get the output (1.1GD.NR). The *Loss Data Term* is relatively low with 0.00172 which is expected because the only thing the algorithm is trying to do is to reverse the previous blurring step. Therefore an image is created which will reproduce the blurred image when the blurring step is applied. On the other hand we get a huge squared distance difference between the deblurred and the original image with 54.64424. This is because the image contains many artifacts and the previous white parts are now in a gray shade. The artifacts (rastering of the whole image) can be traced back to the used kernel.

1.1.b Impact of the λ value for Gaussian Prior and Anisotropic Total Variation

Gaussian Prior Regularization Term

When using the *Gaussian Prior Regularization Term* the artifacts that were present without any regularization term are less pronounced and the shapes do have a more uniform color (the gradients between each pixel within a shape are lessened). Because of the introduction of a regularization term the *Loss Data Term* value is increasing proportionally witht the λ value. The best approximation of the deblurred image is reached with a λ of 0.002 (1.2GD.GP.2). With 31.1901 this difference is still pretty high and the range of the regularization term's weight should not be too high because otherwise it will worsen the deblurring of the image.

As the outputs of the algorithm (1.2GD.GP.1 to 1.2GD.GP.6) show, the regularization term is reducing the gradient of adjacent pixels, such that the pixels which are of approximately the same color are now more uniform. But when the λ is too high the effect of the deblurring is reversed and the initial blurred image can be identified in the deblurred image.

### Anisotropic Total Variation Regularization Term

Using the *Anisotropic Total Variation Regularization Term* produces the best result for the Gradient Descent method. Again it is uniforming the values of the pixels that are approximately the same, by lessening the gradient between them. For a $\lambda = 0.0075$ (1.3GD.ATV.3) the difference to the original image is only 5.20552. Also the Loss Data Term is very small with 0.02146. Also a λ value of 0.01 still produces a very good deblurring result.

A higher λ on the other hand will quickly lead to worse results in which a noise is introduced, "ruining" the previous steps in order to deblur the image. Also the *Loss Data Term* is "exploding" with a value of 3.24555. Therefore a correct λ value is of highest priority for this approach.

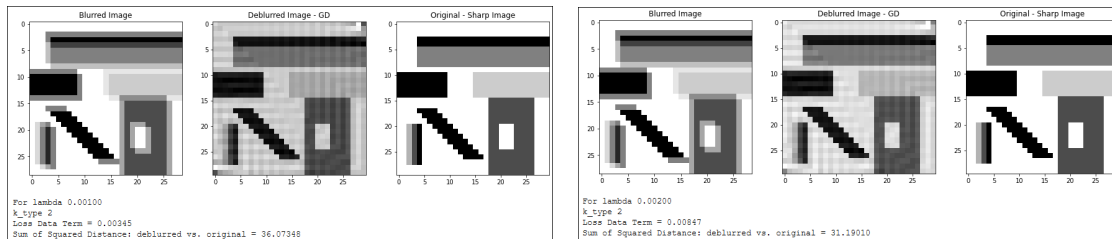## Algorithm Output for Gradient Descent

The following section contains the outputs of the *Gradient Descent Algorithm* from the Jupyter Notebook, which were referenced in previous sections.
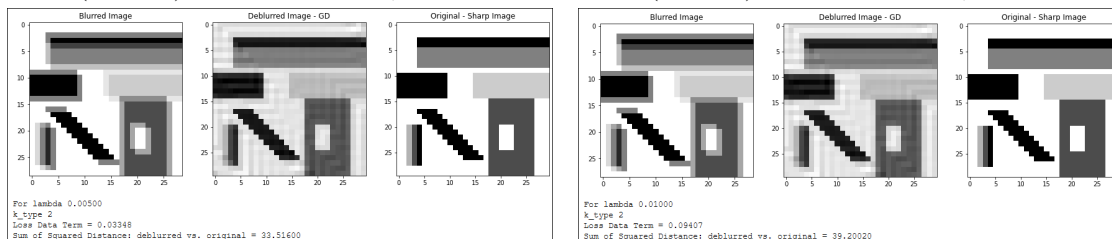
No Regularization
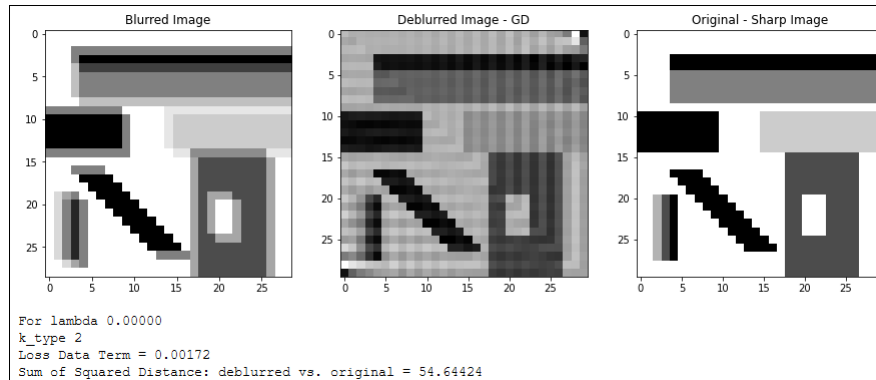
(GD.NR) Gradient Descent - *No Regularization*
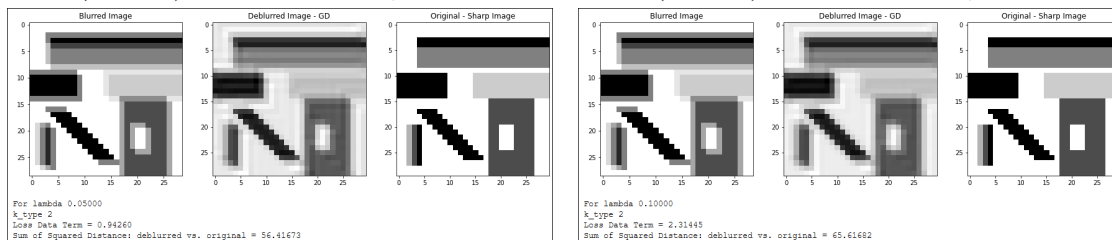
Gaussian Prior

(GD.GP.1) GD - *Gaussian Prior*, $\lambda = 0.001$

(GD.GP.2) GD - *Gaussian Prior*, $\lambda = 0.002$

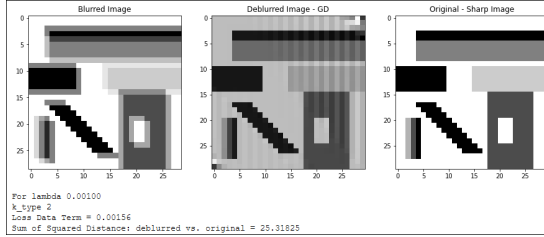(GD.GP.3) GD - *Gaussian Prior*, $\lambda = 0.005$
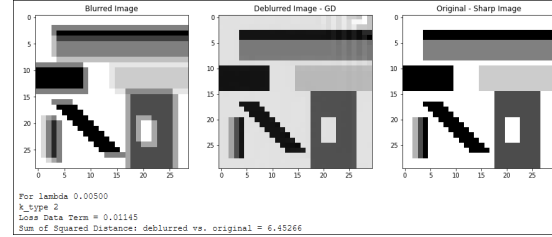

(GD.GP.4) GD - *Gaussian Prior*, $\lambda = 0.01$

(GD.GP.5) GD - *Gaussian Prior*, $\lambda = 0.05$

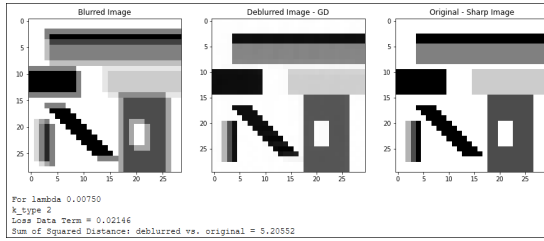(GD.GP.6) GD - *Gaussian Prior*, $\lambda = 0.1$
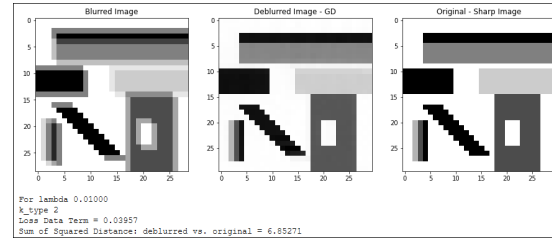
Anisotropic Total Variation

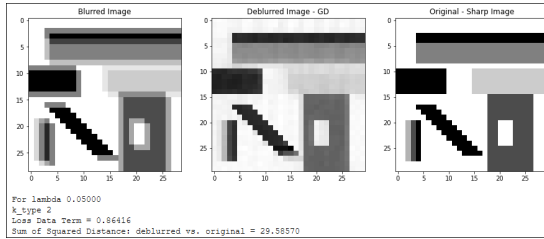(GD.ATV.1) GD - *Anisotropic TV*, $\lambda = 0.001$

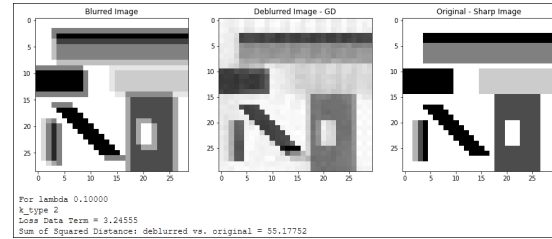(GD.ATV.2) GD - *Anisotropic TV*, $\lambda = 0.005$

(GD.ATV.3) GD - *Anisotropic TV*, $\lambda = 0.0075$

(GD.ATV.4) GD - *Anisotropic TV*, $\lambda = 0.01$

(GD.ATV.5) GD - *Anisotropic TV*, $\lambda = 0.05$

(GD.ATV.6) GD - *Anisotropic TV*, $\lambda = 0.1$

1.2   Linearization + Gauss-Seidel

1.3   Linearization + SOR

1.4   Other Questions

1.4.1   Effect of the λ value

1.4.2   Sum of Squared Destinations