CNT Composite Image Reconstruction

Venelin Martinov
Supervisor: Samuel Heroy

September 2019

1 Introduction

This report shows the work done on image reconstruction for the composite pictured in figure 1. The composite is a mix of nanotubes in a polymer. The pictures are of crystals grown around the nanotubes. Three main problems related to the reconstruction have been considered. The first one is is fitting a suitable analytic function to the two-point correlations derived from the images. This function is later used to normalize the correlations, so that they can be compared between images and at different concentrations of the nanotubes. The second problem is predicting the two-point correlation function of an intermediate concentration of nanotubes using images of higher and lower concentration. The third problem considered is the actual generation of images. For this goal we attempted two techniques. The first method is called simulated annealing. It is an optimization technique which only uses the two-point correlation function and produces a binary image which matches it as close as possible. The second method for image generation we tried is a neural network architecture called a convolutional autoencoder. This method can be used to generate images in the style of the images it trained on using a much smaller initial seed image.

2 Normalization

The main difficulty with comparing different images at the same or different concentration are their varying thresholding behaviours. Some of this can be seen in figures 2, 3 and 4. In order to overcome this, we have adopted a method of normalizing the two-point correlation functions of the images. By dividing the correlation function by a suitable interpolation between the percentage of filled pixels and the square of that we can compare pictures with different densities of black pixels, as well as different concentrations of nanotubes. The rationale for that is that as the range considered approaches zero pixels will be completely dependent and the two-point correlation will just be the density of filled pixels. At very long ranges, pixels will be completely independent and the two-point correlation is the probability of both pixels being filled, which is the square of the density. In figures 2, 3 and 4 the results of normalizing by

the linear interpolation between those two points can be seen. In figure 5 the results of fitting an exponentially decaying function can be seen. This has been used for normalization in the results on simulating the two-point correlation of the intermediate concentration in the next section with an exponent of 5 The exponentially normalized two-point correlation functions can be seen in figure 6. The exponential function seems to be good fit for most of the pictures but some local effects might affect the shape of the correlation function.

3 Approximating the Correlation

The point of this section is to describe the process of interpolating between the information about the .1% pictures and the .6% ones to produce an approximation to the two-point correlation of the .3% pictures. The important property of the normalization used above is that it is reversible and its only parameters are the exponent and the density of black pixels. So once a suitable interpolation has been made between the two concentrations it is enough to multiply it by the function used for normalization to produce a two-point correlation function. The averaged correlations can be seen in figure 7 and the results of the interpolation between the two concentration can be seen in 8. These simulated two-point correlation functions can then be used to generate an image of the desired concentration using the simulated annealing process described below.

4 Simulated Annealing

This is the process used for generating an image from a given two-point correlation function. The main reference for the work on this technique is [3]. The method is based on annealing in metallurgy and was first introduced in [2]. The simulation is initialized with a random binary picture with a given number of black pixels. A fictitious temperature is set at some value and slowly cooled. At each step two pixels of different colours are chosen and switched with some probability. The probability depends on the "energy" of the new state and the temperature. The energy is the two-norm of the difference between the two-point correlation of the image and the targeted correlation. If the new image is "closer" to the targeted function, then the pixels are always switched. If it is further away, then the pixels are switched with some probability, which reaches zero when the temperature reaches zero. The main idea is that "bad" switches are made while the temperature is high and the algorithm slowly transitions to a greedy one as the temperature is lowered.

Some attempts were made to reproduce the picture in figure 1. The image was binarized at several different thresholds. A comparison of the pixel intensities and portion of black pixels at different thresholds can be seen in figure 9. The two-point correlation function used in the simulation was sampled along the horizontal and vertical directions with no boundary conditions. For the simulations we used 300 by 300 pixel parts of the image. The initial temperature

Table 1: Architecture of the Convolutional Autoencoder.

Layer	Number of Filters	Size of Filters
1st	64	6x6
2nd	32	6x6
3rd	16	9x9
4th	8	9x9
5th	1	2x2

was set to 10^{-6}. The simulation was run for 2.4 million iterations and the temperature was lowered linearly to zero for the first 2 million steps. The simulated image was then run through a Gaussian filter with standard deviation 1 to get rid of small artifacts. The filter affects the number of black pixels minimally. Some of the results can be seen in figure 10.

5 Convolutional Autoencoder

The main reference for this method is [1]. We used convolutional autoencoders, since they seem to be a more modern approach to the problem, in contrast to the deep belief networks used in the original paper, which are also somewhat harder to implement. The main goal is to reconstruct similar images using much fewer parameters, potentially from a random reduced image. Only the work on reducing the original images and reproducing them is included, but the same architecture can be used for random image generation.

I have used 256 by 256 pixel subsets of the original image to train the neural network, using 20% of the images for validation. The parameters of the encoder can be seen in table 1, with the decoder having the same architecture in reverse. The network has five convolutional layers with a max-pooling layer after each, which reduces the size of the image in half. The network shrinks the image to 8 by 8 pixels and then attempts to reconstruct it. Results of the method can be seen in figure 11.

6 Conclusion

The results of the simulated annealing process can be extended to larger pictures and improved by more iterations. The simulated images might be smoothed afterwards using some off-the-shelf algorithm to get a non-binary image if desired.

The autoencoder images are somewhat blurry, which is probably unavoidable, due to the fact that the reconstruction is based on an 8 by 8 representation of the original image. The results could be improved by additional training data. Moreover a related network architecture might yield clearer images - a generative-adversarial network, although it is probably going to require more training data.

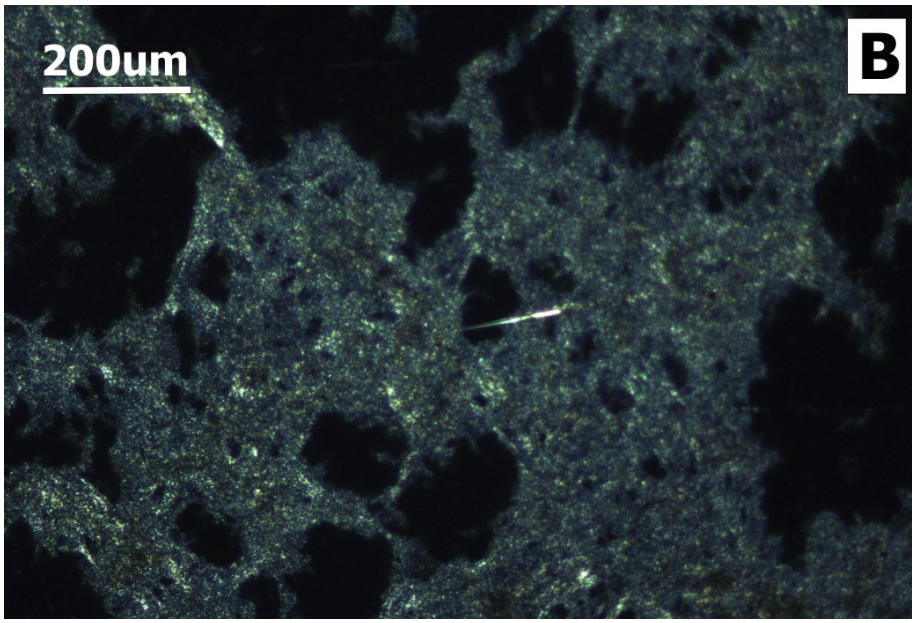

Figure 1: The Image

References

[1] Ruijin Cang, Yaopengxiao Xu, Shaohua Chen, Yongming Liu, Yang Jiao, and Yi Ren. Microstructure representation and reconstruction of heterogeneous materials via deep belief network for computational material design. *Journal of Mechanical Design*, 139, 12 2016.

[2] Mark D. Rintoul and Salvatore Torquato. Reconstruction of the structure of dispersions. *Journal of Colloid and Interface Science*, 186(2):467 – 476, 1997.

[3] N. Sheehan and S. Torquato. Generating microstructures with specified correlation functions. *Journal of Applied Physics*, 89(1):53–60, 2001.

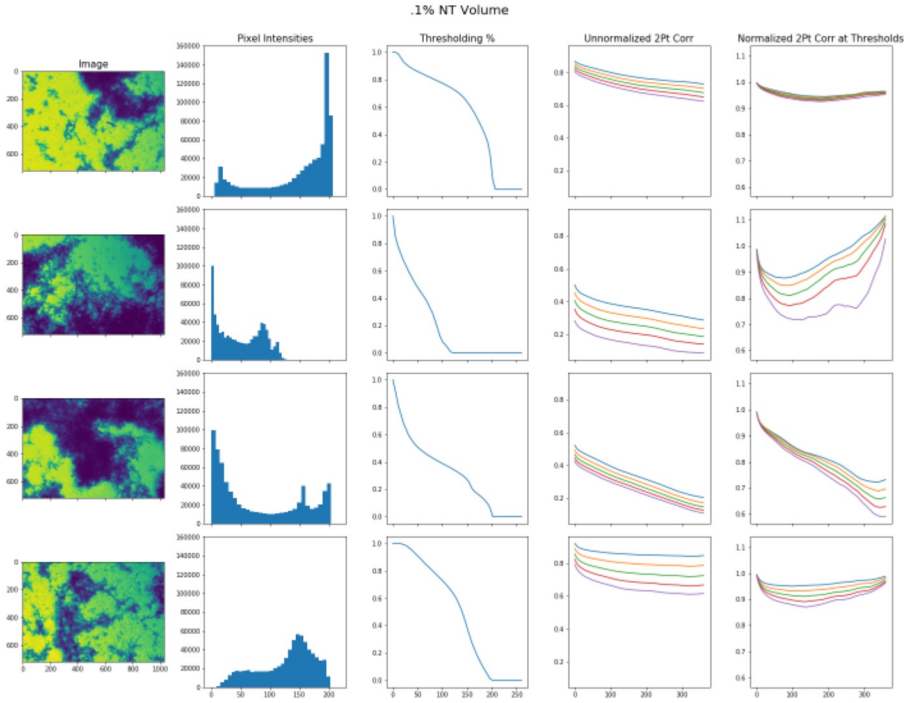

Figure 2: The pictures used as well as plots of some of their properties. Starting on the left: (1) is the grayscale picture. (2) is a histogram of the intensities of the pixels on a scale to 256. (3) is the percentage of pixels above a given threshold. (4) are the two-point correlations at five thresholds equally spaced between 45 and 80. The values are chosen so that the functions behave nicely and exhibit interesting behaviour. (5) are the linearly normalized two-point correlations at the same thresholds.

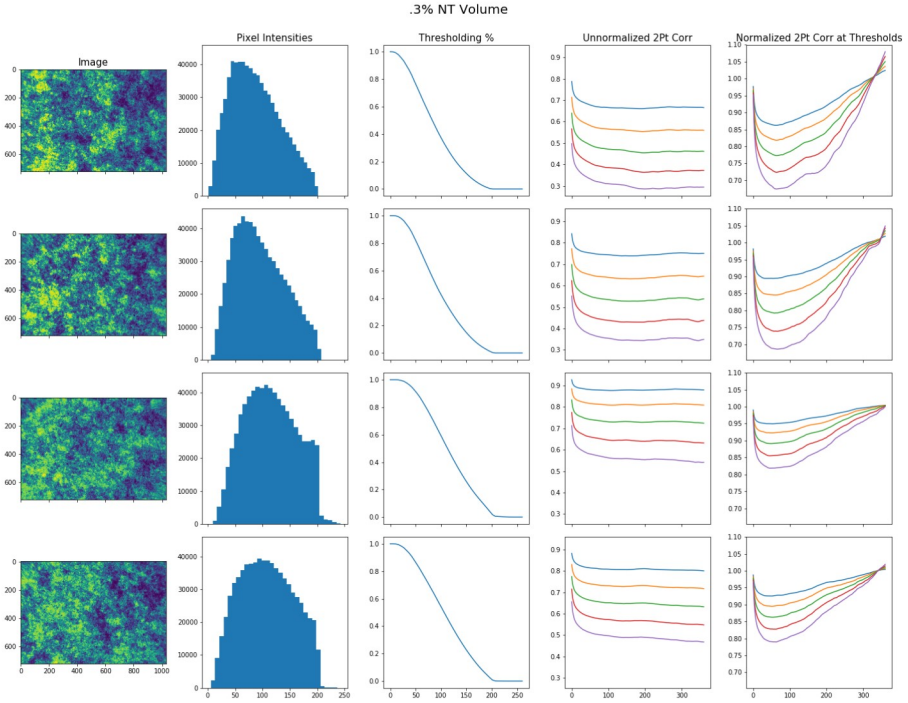

Figure 3: The pictures used as well as plots of some of their properties. Starting on the left: (1) is the grayscale picture. (2) is a histogram of the intensities of the pixels on a scale to 256. (3) is the percentage of pixels above a given threshold. (4) are the two-point correlations at five thresholds equally spaced between 45 and 80. The values are chosen so that the functions behave nicely and exhibit interesting behaviour. (5) are the linearly normalized two-point correlations at the same thresholds.

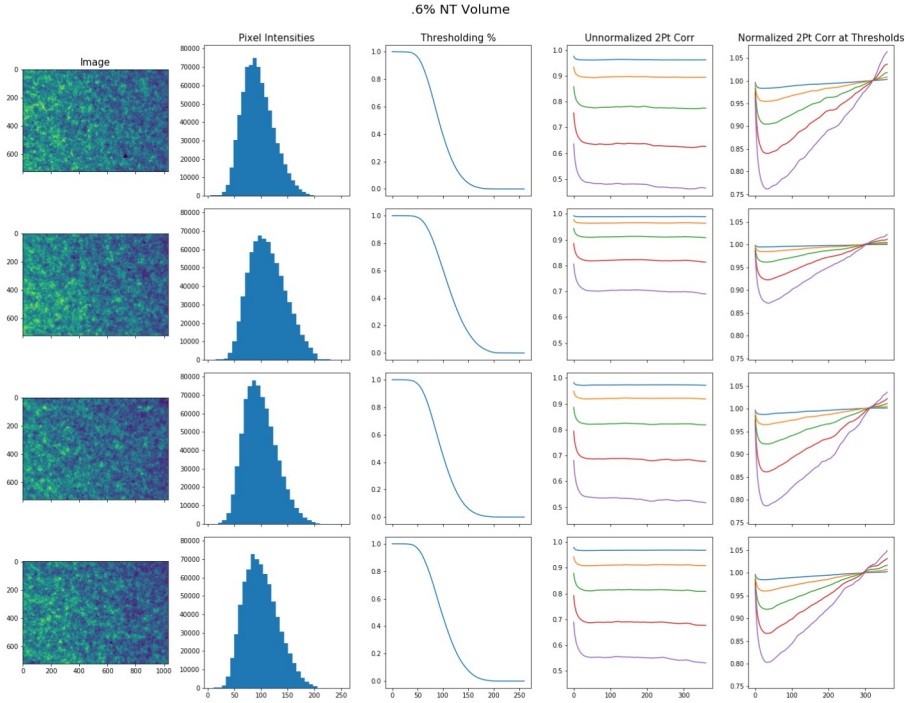

Figure 4: The pictures used as well as plots of some of their properties. Starting on the left: (1) is the grayscale picture. (2) is a histogram of the intensities of the pixels on a scale to 256. (3) is the percentage of pixels above a given threshold. (4) are the two-point correlations at five thresholds equally spaced between 45 and 80. The values are chosen so that the functions behave nicely and exhibit interesting behaviour. (5) are the linearly normalized two-point correlations at the same thresholds.

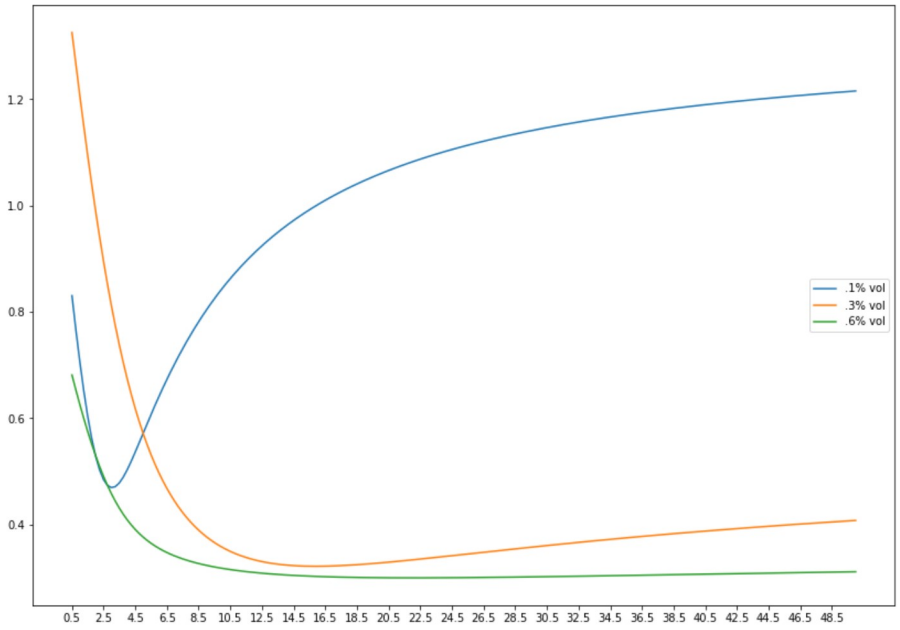

Figure 5: Measuring the errors of the two-point correlations of the three different concentrations with an exponentially decaying interpolation between the density of filled squares and its square. The errors are the Euclidean distances between the vectors and are then averaged for each concentration over the different pictures and thresholds. The function used for normalization is $\frac{e^{\lambda x}(d-d^2)}{e^{\lambda}-1} + d^2$, where λ is the coefficient which is varied on the horizontal axis, x ranges from 1 to 0 and d is the density of the black pixels in the particular thresholded picture.

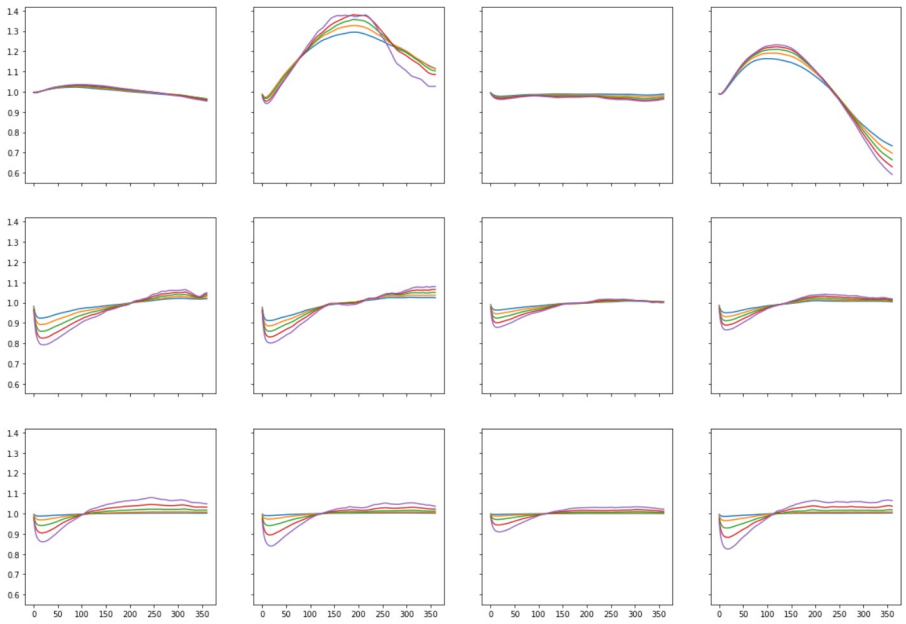

Figure 6: The two-point correlation functions of the different pictures at different thresholds normalized by an exponentially decaying function as described in figure 5. The first row are the functions for the .1% volume nanotube pictures, the second is for the .3% and the third is for the .6%. The colours are for the different thresholds considered as described in figure 2

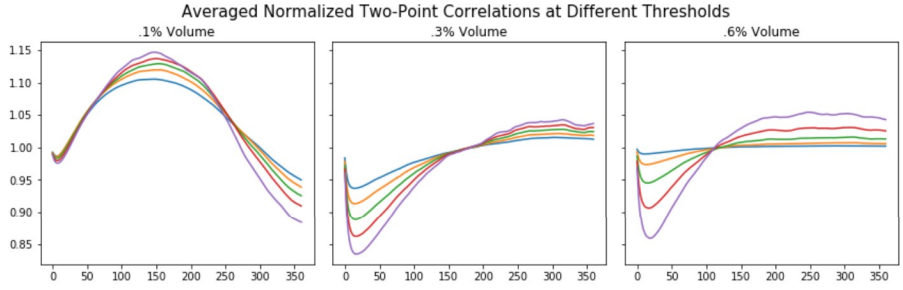

Figure 7: The averaged normalized two-point correlations for each concentration. The colours are the different thresholds as described in figure 2.

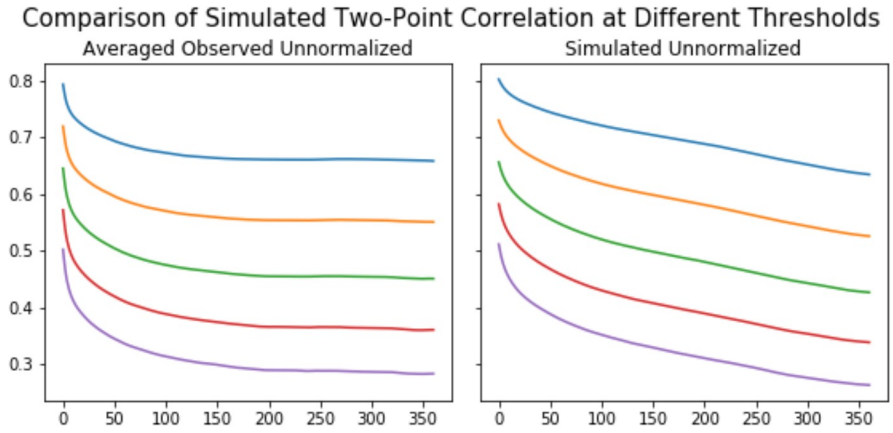

Figure 8: The averaged unnormalized two-point correlations of the .3% pictures versus the interpolated unnormalized results. The colours are the different thresholds as described in figure 2.

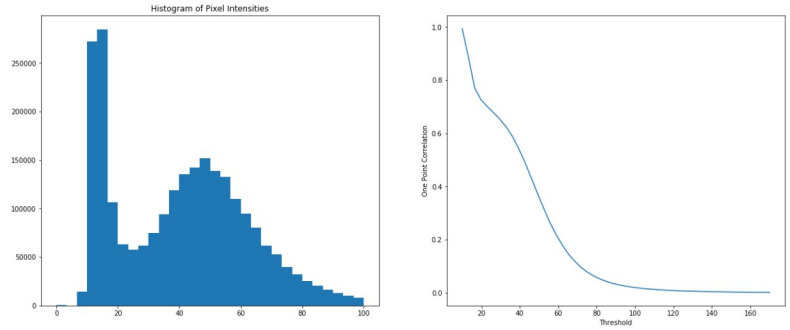

Figure 9: Comparison of the different threshold for binarization. On the left is the histogram of the intensities of the pixels in the original image. On the right is a plot of the percentage of pixels remaining versus the threshold.

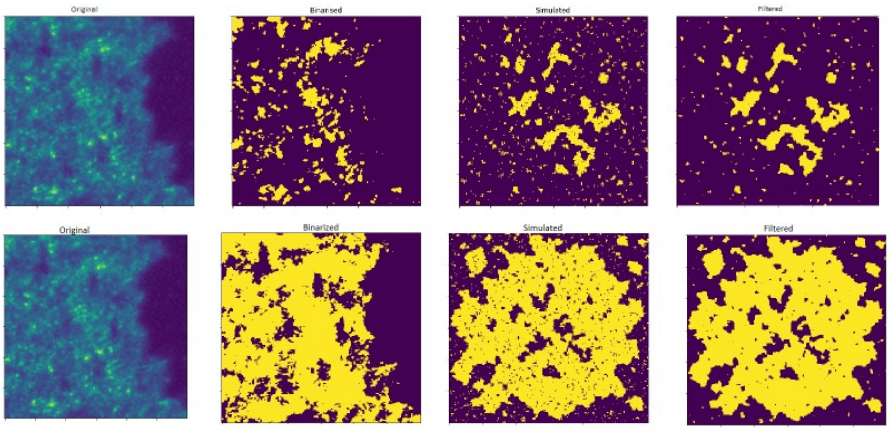

Figure 10: The results of the simulated annealing at two different thresholds. The two-point correlation functions of the binarized and simulated images are virtually identical. Starting of the left: (1) is the original image. (2) is the binarized image at a given threshold. (3) is the result of the annealing process. (4) is the result after applying the Gaussian filter. The filtered image has around 3% less black pixels.

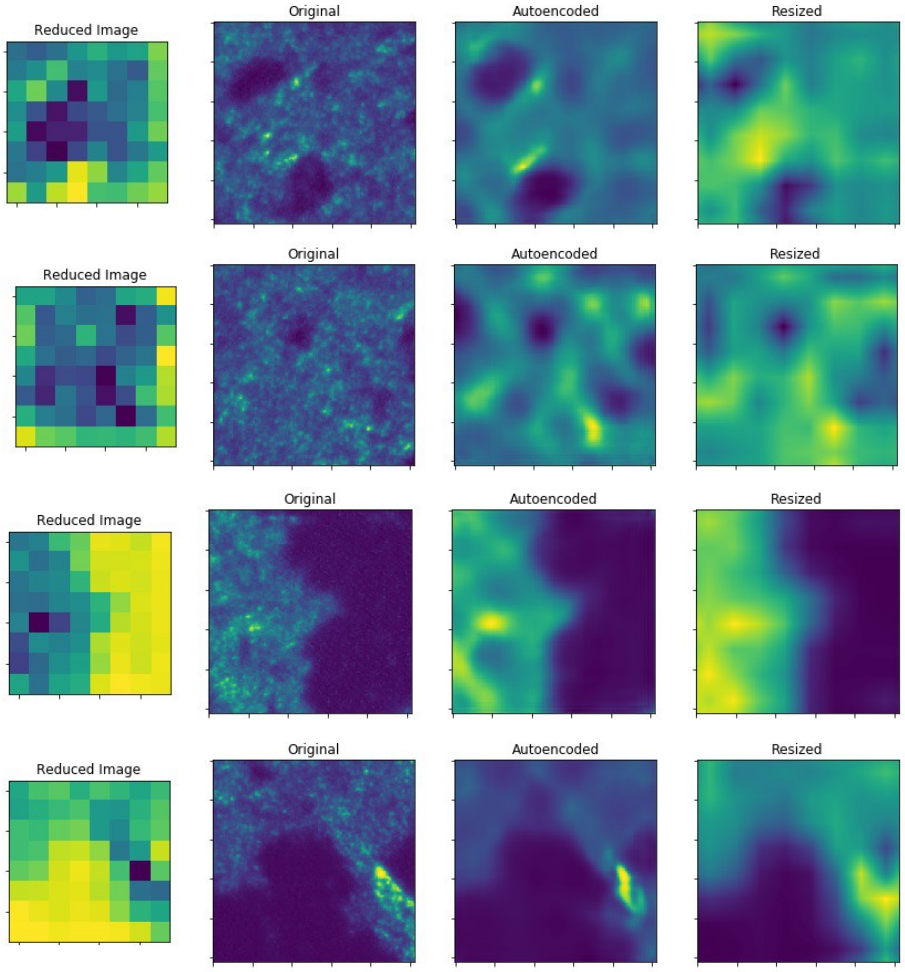

Figure 11: The results of the autoencoder. Starting on the left: (1) is the shrunk image. Note that the colours are inverted. (2) is the original image which was used as input. It is a 256x256 pixel subset of 1. (3) is the reconstructed image from the autoencoder, which only uses the 8x8 image as input. (4) is the result of shrinking the original to 8x8 and then interpolating between the pixels to get a 256x256 image using a standard image interpolation algorithm for comparison.