Decompressing Corrupted Images

By Alanna Manfredini

Background – Summary of Theory and Impact

In modern society, millions of images, often in the form of videos, are transmitted across the internet and stored every second. When an image is stored on a computer, it must be compressed and then decompressed to avoid a single image taking up a ubiquitous amount of storage space. When said image is compressed, the maximum redundant data in the image are dropped to minimise the storage consumed (Sha). Thus when the image is decompressed it must have a mathematical technique to restore the image with the same quality as the original. On the other hand, images are often corrupted by physical means: an old photograph may have been faded from the sun. Both of these situations require an algorithm to recognise the missing data in the image and fill in the missing spaces with the best approximation of what data used to be there. The method used throughout this project uses Discrete Cosine Transforms and LASSO regression.

Discrete Cosine Transforms

An image can be divided into a number of *dxd* sized chips which are comprised of superimposed cosine waves with different weightings. The transformation of the image into a set of coefficients, one corresponding to the weighting of each wave, is called a Discrete Cosine Transform (DCT) (Tan and Gan). This process is done when compressing images so only the string of coefficients has to be sent rather than every pixel value: the size is thus reduced to the square root of the original number. Additionally, since many of the coefficients are zero or can be approximated as zero without any visual loss in clarity, the list of coefficients can be further compressed through Huffman encoding (Weik).

LASSO Regression

When reconstructing a corrupted image, the weighting of the DCT cosine waves that would achieve the intensity of each sensed pixel is found. Thus, using the intensity of the given pixels, LASSO regression must be performed to find the coefficients that match said intensity. When finding a model for datapoints regression is performed to penalise the amplification of a parameter and avoid overfitting. The loss function is $||y - \Phi\Theta||^2 + \lambda||\Theta||^2$ where λ is the regularisation parameter and controls the "strictness" of the model. LASSO regression stands for least absolute shrinkage and selection operator. In this regression the L-1 norm is specifically chosen, to compensate for sparse datapoints (Marc Peter Deisenroth et al.). Images are considered sparse because when corrupted many of the pixels (datapoints) are "off" and therefore provide minimal information (Di Gesù). Additionally, images have similar colours near each other which further reduces the complexity of the data, and, since images can be represented through wavelet transforms, the recursive nature of the wavelet drives most of the coefficients to zero (Dimililer and Kavalcıoğlu). Throughout this project it is possible to observe the impact that the sparsity of an image has on the convergence of models and the regularisation parameter. After the image is reconstructed using LASSO regression, the image is filtered with median filtering.

Median Filtering

Median filtering is used to reduce gaussian and random noise. Throughout this process, the image was reconstructed block by block and then tessellated back together. This method meant that adjoining blocks had clearly different colour without transition between them. It works by running through each pixel and replaces it with the median of its neighbouring pixels. (Kaufmann). It therefore this removes any outliers includeing noise. (Huang et al.) Different shapes of neighbouring windows are used depending on the amount of filtering needed or the direction of filtering (Pitas and Venetsanopoulos). An increase in size of the filter window smooths the final image and different shaped windows are more appropriate for different noise distributions, for example images with smaller percentages of impact noise added are better approximated by a Z-shaped window than a square window (Appiah et al.). A median filter is chosen, because it is a non-linear filter. Median filters aim to reduce outliners, whereas other filters such as a mean filter merely smooths an image or a high pass filter which cuts off all frequencies lower than a certain value and thus doesn't deal with upper limit outliers or pixels which are supposed to have a low frequency. The mean squared error was found for both the filtered and unfiltered reconstructed images to highlight the improvement smoothing can have on reducing noise in an image.

Original Images

Left: Boat Image (200x192pxl)

Right: Nature Image

(512x640 pxl)

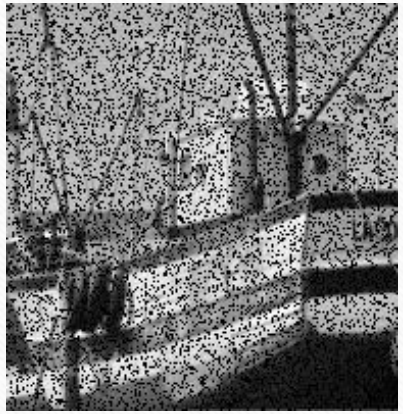
These images were loaded from a .bmp file and visualised with a PIL package

Corrupted Boat Images

S64

The first step of the project was to corrupt the images so that they could then be reconstructed. Each block of an image was taken, rasterised and then a certain number (S) of random pixels were removed. Each block was then reconstructed into a full image. Pictured below is the boat image that has been corrupted by removing a specific number of pixels in each block. Moving from left to right the number of pixels removed are 14, 24, 34, 44 and 54 leaving 50, 40, 30, 20 and 10 sensed pixels respectively. It is interesting to note that as a human it is easy to distinguish the original image despite much of it being missing; this highlights that humans are able to distinguish trends across entire images rather than just blocks, something that is not explored in this project.

S50

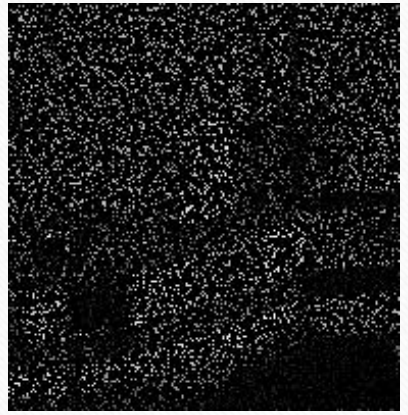


S40

S30

S20

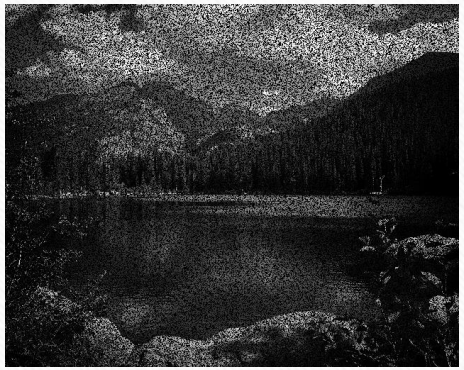
S10



Corrupted Nature Images

S256

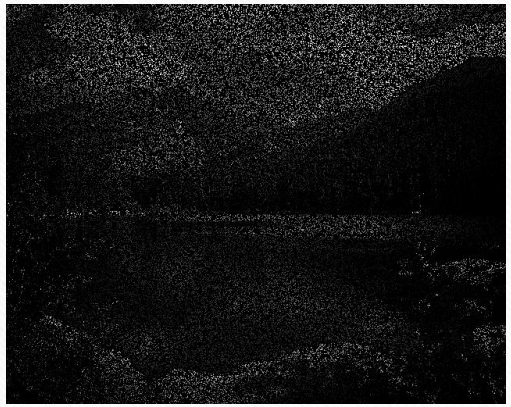
As was explained on the previous slide, the images were split into blocks, each block was corrupted and then the image was reconstructed. The ”S” number represents the number of sensed pixels remaining



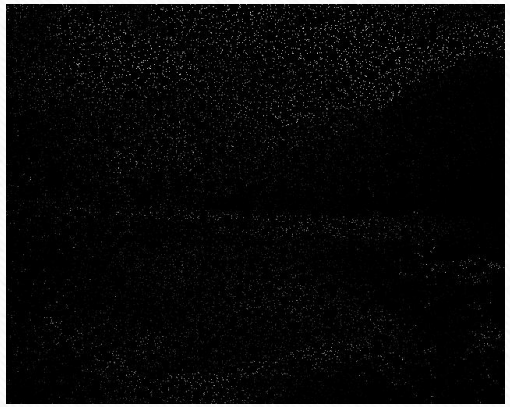
S50

S30

S150



S100

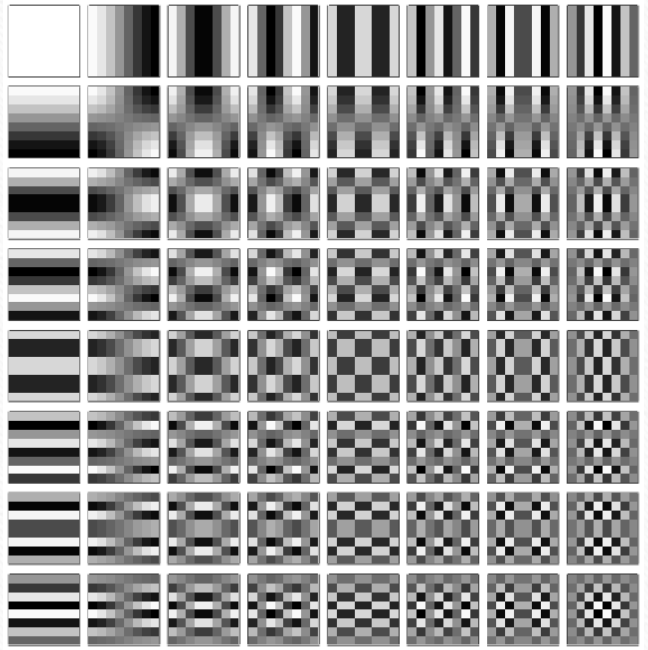


S10

Basis Vector Images

Basis Vector Images form the compression and decompression of many image types including jpegs (Sha). An image can be divided into sets of square blocks which can then be tessellated back together. Every block of an image is the result of superimposing a set of cosine waves, pictured to the right, on top of each other. Each of these waves is weighted to make the unique block in the image (Tan and Gan).

In the image to the right the block size is 8x8. This is the size of each wave image and also the number of possible waves.

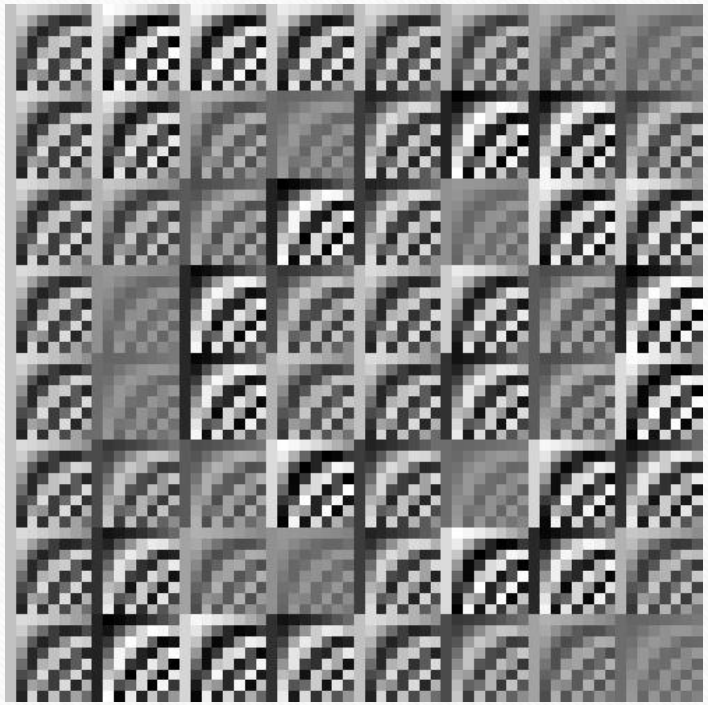


Cosine wave group ("DCT")

Boat Basis Vector Images

This basis vector images was produced by rasterizing each of the wave images in the previous slide. Each column corresponds to a different wave image moving vertically down the group of wave images. Every row of the basis vector image corresponds to a pixel in a block. Therefore, by using all the pixels in a block as datapoints the approximate weighting of each wave can be found.

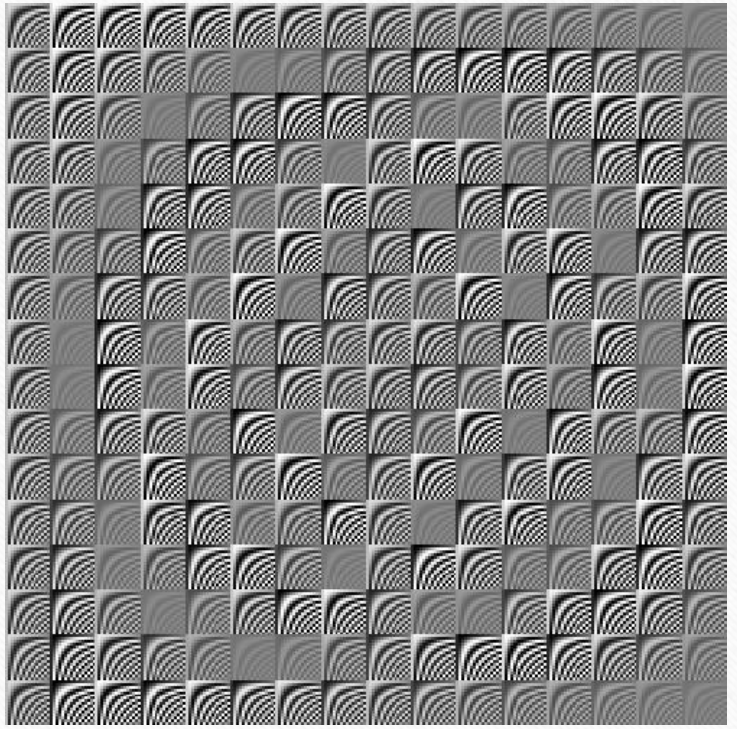
When reconstructing the image, these weighted coefficients can be applied to every wave and they can be superimposed which results in the rest of each 8x8 block being approximated. The act of breaking an image into coefficients of cosine waves is called the Discrete Cosine Transform.



Nature Basis Vector Images

As was described on the previous slide, the cosine waves can be rasterized and used to form a basis vector image which can in turn be used to estimate the Discrete Cosine Transform coefficients.

The image on this slide is considerably larger than the previous slide because each block is 16x16 pixels. Therefore every block must be made of 256 different cosine waves layered on top of each other. This image is 256 pixels that correspond to each pixel in the block x 256 different cosine waves.



Undetermined Linear Systems

Undetermined linear systems are especially important to solve in cases of rank-deficiency, situations which are evident in sparse image reconstruction. They must be dealt with by requiring "smoothness" of a function. Additionally, a term must be added to account for ill-determined inversed whilst still ensuring smoothness, the regularisation parameter. Most commonly, ill-posed problems are in situations where datapoints in a vector are function values of a function (Neumaier).

LASSO Regression

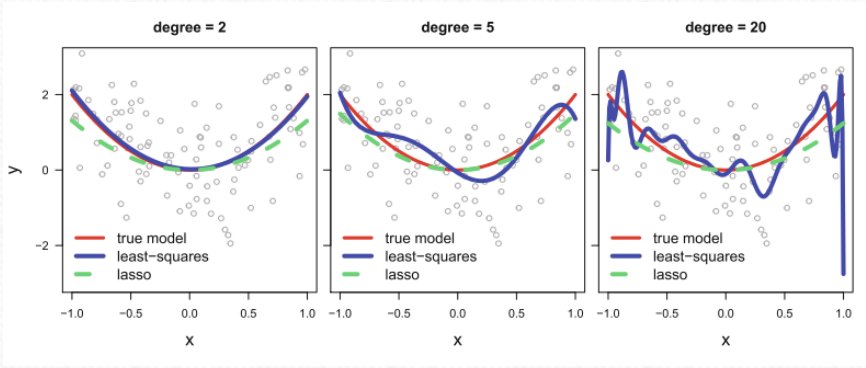
Reconstructing a corrupted image requires creating a model for many coefficients from few datapoints. LASSO regularisation is a commonly utilised method for high dimensional problems, problems where the number of unknown parameters is higher than the number of observations. (Gauraha) In this case LASSO is chosen as the regularisation approach because images have d^2 pixels that must be estimated by d^2 cosine functions, but there may be as few as 10 pixels to inform the creation of the model.

A norm is a function which assigns a length to each vector. Different norms can be categorised in different ways such as the Euclidian norm where the length $= \sqrt{\Sigma x_i^2}$ or the l_1, Manhattan norm, where length $= \Sigma|x_i|$ (Marc Peter Deisenroth et al.).

LASSO is characterised by the l_1 norm and is used because manhattan distances ensure that the fuction is convex and produces sparse solutions (Lederer).

Regularisation is the addition of an extra term into the formula to produce $\frac{1}{2N}||y - Xw||^2 + \alpha||w||$ where α or λ acts to restrain the magnitude of the coefficients in the model and minimise variance. This penalty term is a way to compromise between the complexity of a solution and the accuracy of a model. In an image where there is only a few datapoints, it is critical that the model does not overfit to these points to cause massive variation in intensity of the pixels that are being reconstructed. Generally, images have many pixels of the same colour next to each other so any overfitting can make an image appear very noisy (Marc Peter Deisenroth et al.).

Figure 1: Visualisation of Overfitting without LASSO (Lederer)

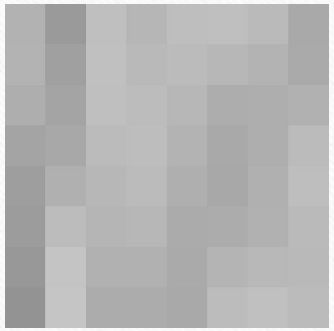


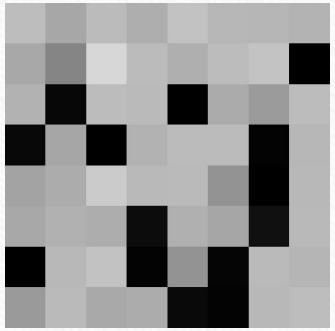
LASSO – Regularisation Constant

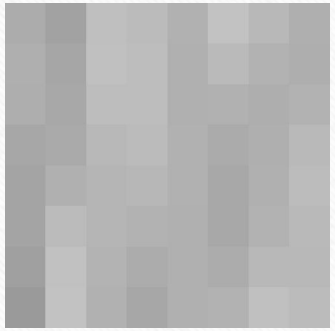
When reconstructing images, the vector basis matrix is cropped to leave only the rows corresponding to the retained pixels. Thus, we are left with a non-invertible matrix. The equation $D = A\kappa$ can then be solved with LASSO Regression in the form $||A\kappa - D|| + \lambda|\kappa|$ where A is the cropped basis vector matrix, D is the sensed pixels and κ is the Discrete Cosine Transform coefficients. This equation was previously stated as $\frac{1}{2N}||y - Xw||^2 + \alpha||w||$.

An important thing to note, however, is that A, the cropped basis vector matrix has a column which has all of its components with equal magnitude. Since the intercept of a regression is already accounted for in the development of the model, the constant column could dominate the regression and affect the convergence of the model. Therefore the constant column is cut from the basis vector matrix and the coefficient corresponding to that cosine wave is determined from the intercept of the model.

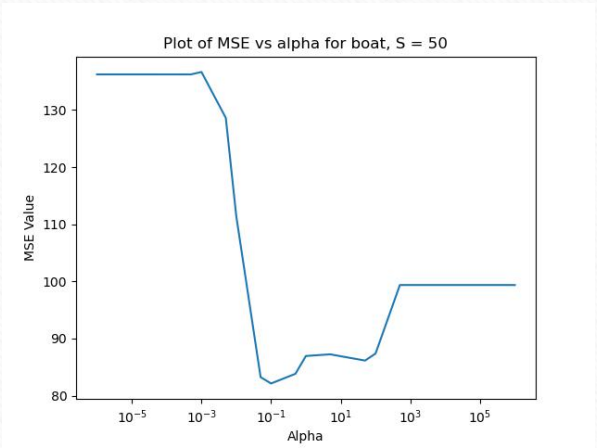
Boat Block Reconstruction

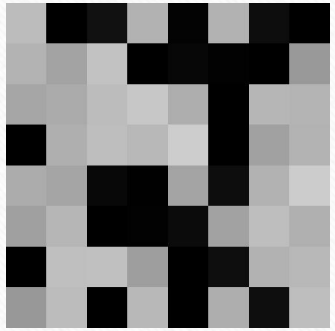


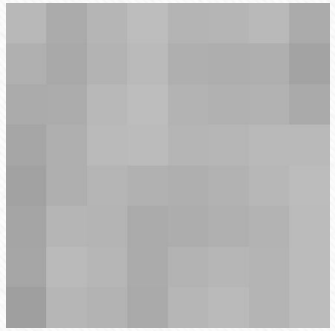




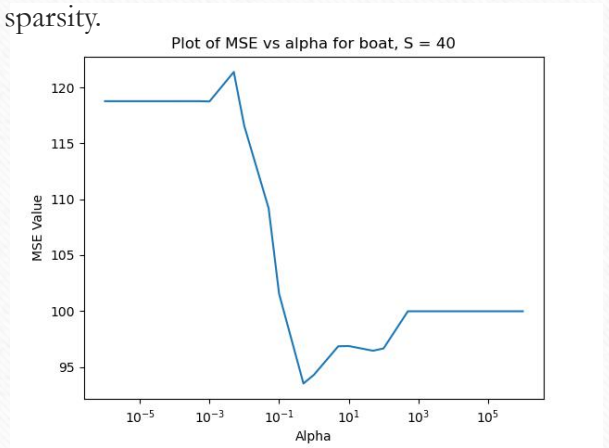
To reconstruct the whole image, each corrupted block was reconstructed with LASSO regularisation. When trialling different alpha values it was determined that $\alpha = 0.1$ had the best MSE for this block. This block, chosen by a formula related to my name, does not have much colour diversity.





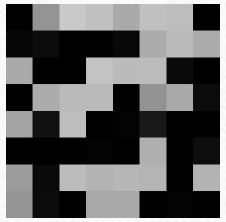


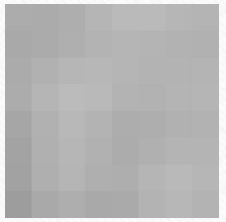
As the sparsity in this image increases, so does the optimal value for alpha. The optimal value for S = 40 is $\alpha = 0.5$. Both of these images have flat lines on the left hand side. This can be attributed to lack of convergence for these values of alpha. A small alpha is equivalent to L1 regression and thus does not converge with sparsity.

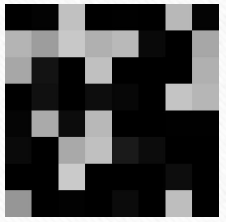


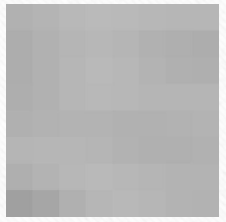
Boat Block Reconstruction

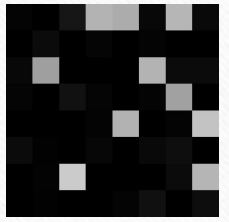
As the sparsity increases there is less and less diversity in the pixel colour across the block. In the last image, S = 10, there is no change in alpha, because it did not converge for any of the values. The reason that it did not converge is because there is minimal change over the pixel values and thus the regression is trying to drive all of the coefficients to zero. Thus alpha does not have a significant impact on the regression and the model is independent of alpha.

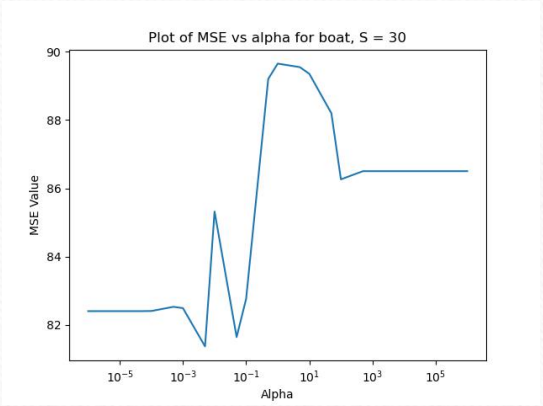


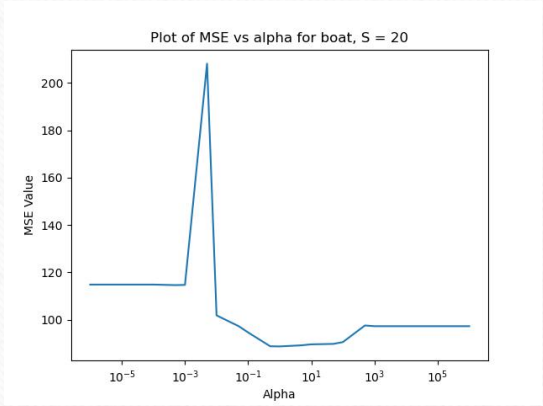


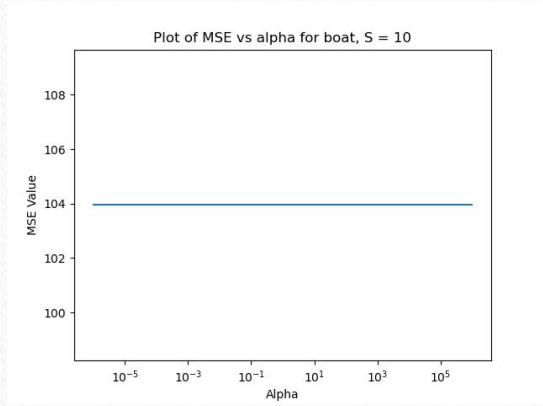






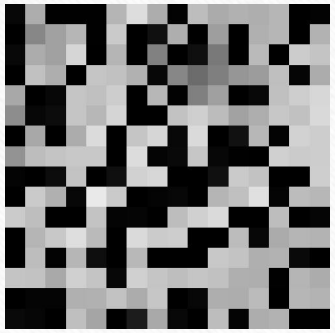


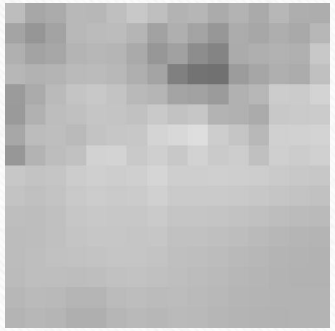




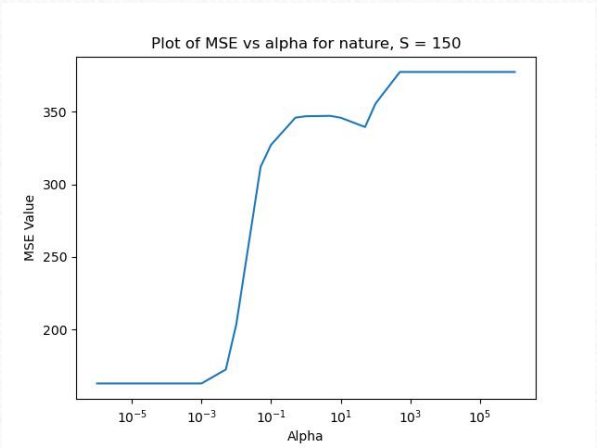
Nature Block Reconstruction

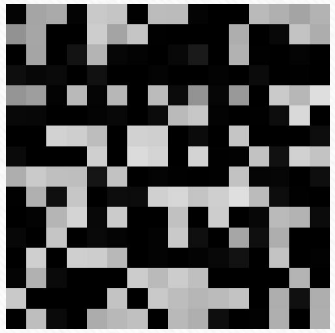
S256



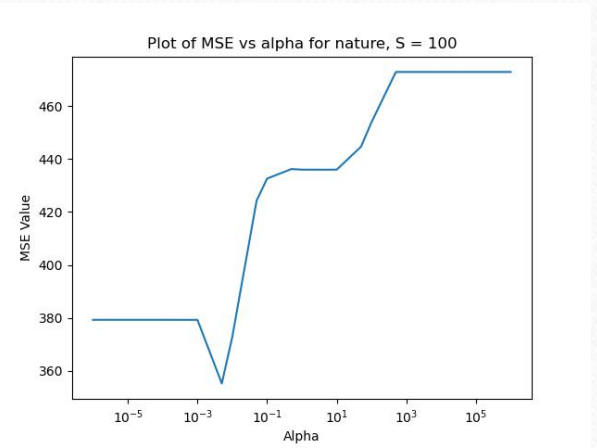


As was done with the boat blocks, the Nature blocks were corrupted and reconstructed individually. The alpha value was similarly approximately 0.1 for the best MSE. It can be observed in the plot below that the model failed to converge with the smaller alpha values.

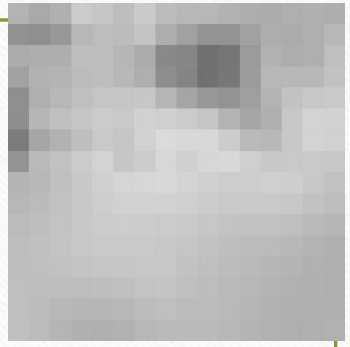




This set of models had a much more clear alpha which minimised the MSE. It is interesting to note that the dark spot in the block is not in this reconstruction because the randomised corruption has cut out pixels in that area.

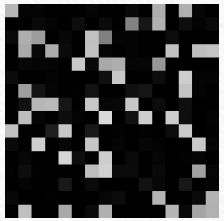


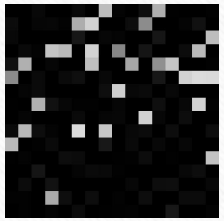
Nature Block Reconstruction

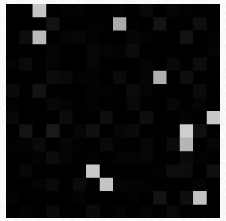


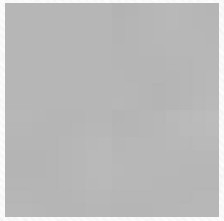
S256

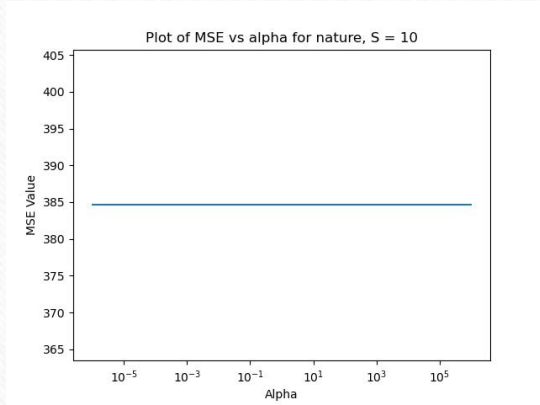
As was mentioned on the previous slide, the dark spot appears in some of the reconstructions (S=30), but does not appear in a reconstruction which theoretically has more information (S=40). This highlights that to find true analytical values of alpha and MSE the corruption and reconstruction must be run multiple times to avoid cutting out any crucial data. Similarly to the Boat Block Reconstruction, S = 10 did not converge and appeared to almost be a flat grey square. There is less diversity in this reconstruction that the boat S=10 because there is proportionally less information : 10/64 vs 10/256.







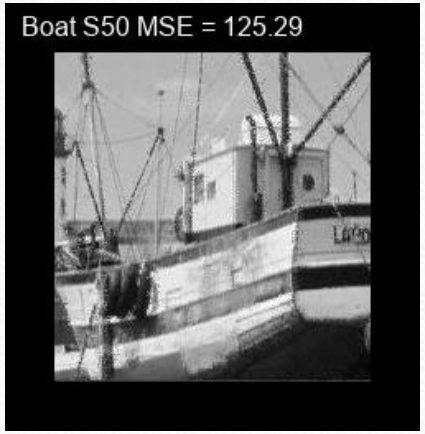


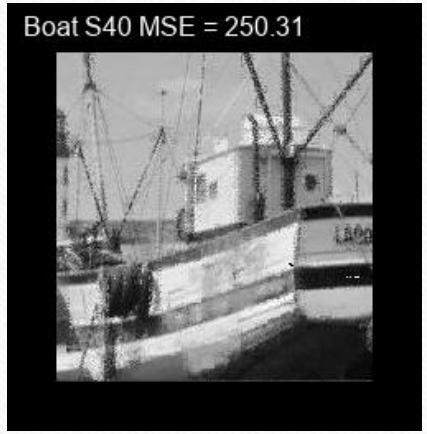


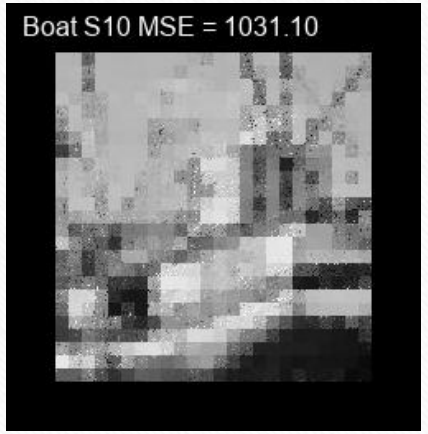
Reconstructed Boat

S64

The method which was used to reconstruct the individual blocks in the previous slide was used to reconstruct an entire image. The corrupted boat images that were attached earlier were split into 8pixel x 8 pixel blocks and then each block was reconstructed individually. It is possible to see this in the S10 image, because there are very defined blocks that show the general trend of the sampled pixels in that location. Visually and numerically the error of each image increases as the number of sampled pixels increases. This was expected, since there is less information to do the regression with. It is interesting to note that the first image is not perfect. This is interesting, because when images are split into their cosine waves, there is no noise and yet the regression could not find the perfect model. The error in the image increases significantly when there is less information. The Mean Squared Error for S10 is almost twice that of S20 whereas this trend is not evident elsewhere.







Introduction to Median Filters

In the reconstructed images pictured on the previous slide, the delineations between each individual reconstructed block is very obvious. Thus, filtering is used to improve image quality after applying LASSO and smooth the transition between the edges of each block. A median filter takes the center pixel of group of a specified size and replaces it with the median of all the other pixels in the group. This is a non-linear filter and therefore works very well at preserving edges unlike other filters such as a mean filter (Banerjee et al.). Median filters effectively reduce noise because noise presents itself as outliers and thus will never be the median of the group of reconstructed images. Additionally, edge preservation is crucial for human interpretation of images.

Unfortunately, if an image is already highly detailed, the filter will remove the finer details such as small lines. In the filter used throughout this project, the size of 3x3 was used, which means any line of width 1 would be removed (Davies). A median filter is still selected over a type of linear filter because linear filters are highly influenced by random noise and do not preserve abrupt changes. This is important for image reconstruction because the noise produced during LASSO is random and does not follow a normal distribution (Fried and George).

Median filtering also has the added advantage that it can be iterated, the image can be re-filtered after the initial filtering to further smooth the image (Kleefeld et al.). As can be observed below, filtering an image without noise and one with noise have very similar outcomes.

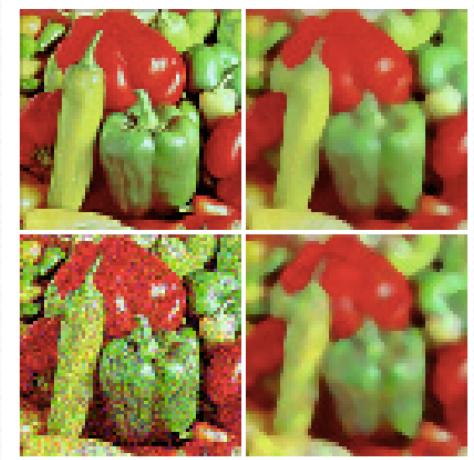


Figure (clockwise from top left): Image, Median Filtered Image, Filtered version of the Image with Added Noise, Image with Added Noise (Kleefeld et al.)

Filtering

Median filtering is chosen over filtering techniques such as low pass or high pass filtering, because it reduces noise whilst preserving crucial image details. Filters work by reducing the disparity between pixel values of neighbouring pixels

Low Pass filtering only allows pixels with an intensity below a certain threshold to be let through. Because of this, any high intensity variation is cut out of the image, which subsequently destroys any image structures present in the original image (Burger and Burge). Due to this, low pass filters are good for removing Gaussian noise, but distort the image if they are used to remove impulse noise in an image (Davies and Netlibrary).

High Pass filtering is utterly inappropriate for use filtering an image because it only lets values above a certain threshold remain. Due to this, it filters out the pixels that are supposed to be in the image and amplifies the noise. High Pass filtering is sometimes used for sharpening an image, however the image must have very minimal noise to be effective (Davies).

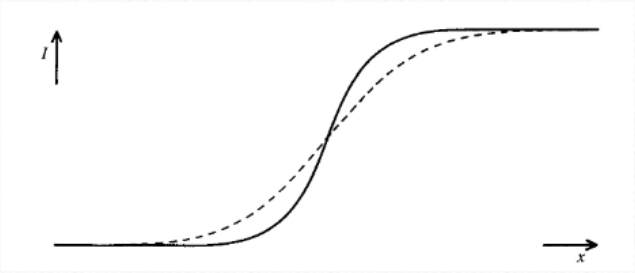


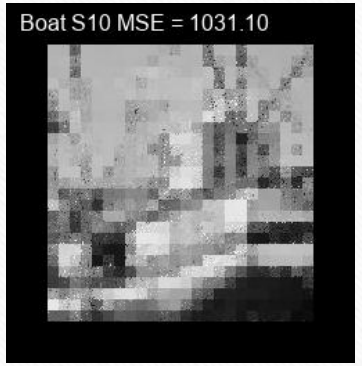
Figure 1: Evidence of blurring during Low-Pass filtering of an image by simple Gaussian convolutions (Davies and Netlibrary).

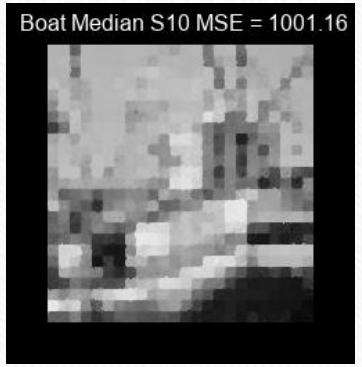
Reconstructed Boat – Median

S64

Each of the reconstructed images that were attached to the previous slides were filtered using a median filter. A median filter acts to smooth out edges between different blocks. It is possible to tell this visually when comparing the unfiltered reconstruction of S10 and the filtered reconstruction of S10 below it. The filtered reconstruction appears to be more blurry than the unfiltered one. As is explored later, the MSE of the median filter is reduced when the edges are softened in the cases with less sensed pixels.

Median filters work well eliminating noise. It is possible to observe "speckling" on the reconstructed image, but this is removed once the median filter is applied on top.





Visualisation of Regularisation Parameter

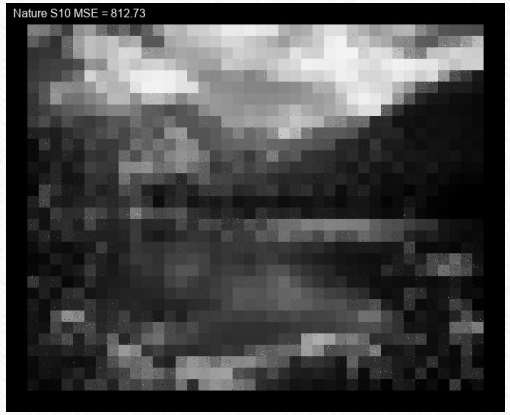
The regularisation parameter of each block is visualised below as a pixel intensity. The intensity varies across the pixel but follows a couple of trends, which can be understood by visually comparing the parameters to the details of the superimposed image. Most of the interesting regularisation parameters are on the sky, masts and cables in the image. These are locations with few details. Since there are few details, the coefficients of a least squares model would be very high and aim to overfit to the few diverse values. Thus the regularisation parameter at these locations is much higher than in other locations so the variance between models is reduced and thus the bias variance trade-off is optimised. The S10 model does not have any variation in model parameter because none of the blocks had enough data for the regression to converge.

S50 S40 S30 S20 S10

Reconstructed Nature

S256

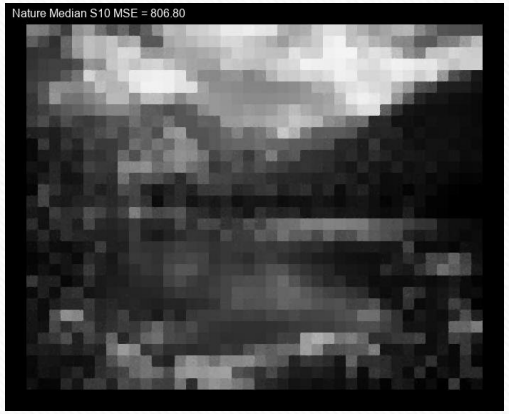
The Nature image was split into blocks, corrupted and reconstructed with various amounts of sensed pixels. The block division is even more obvious when only 10 of 256 pixels remain in the S10 image



Reconstructed Nature - Median

S256

Similarly to the boat image, a median filter of size 3 was applied to the reconstructed images. The median filter smoothed the edges of the blocks. The MSE was much more minimally reduced by the S10 median filter than it was on the boat image.



Visualisation of Regularisation Parameter

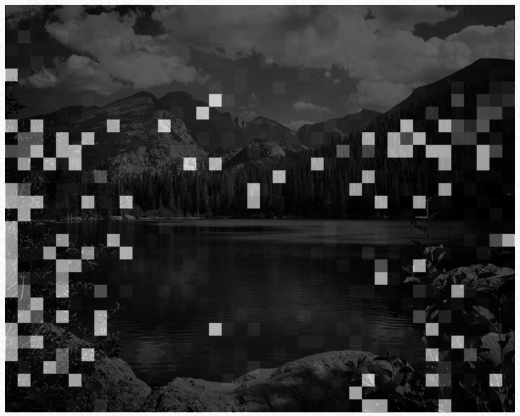
As was seen in the visualisation of the boat's regularisation parameter the more sparse regions have higher regularisation constants. In the nature this is the foliage on the trees and the rocks. This is especially evident in S=50 and S=30.

S50

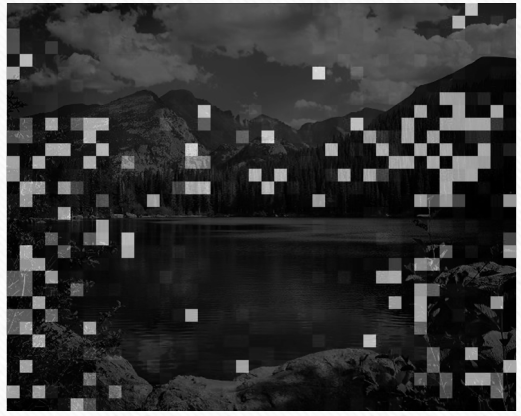
S30

S10

S150



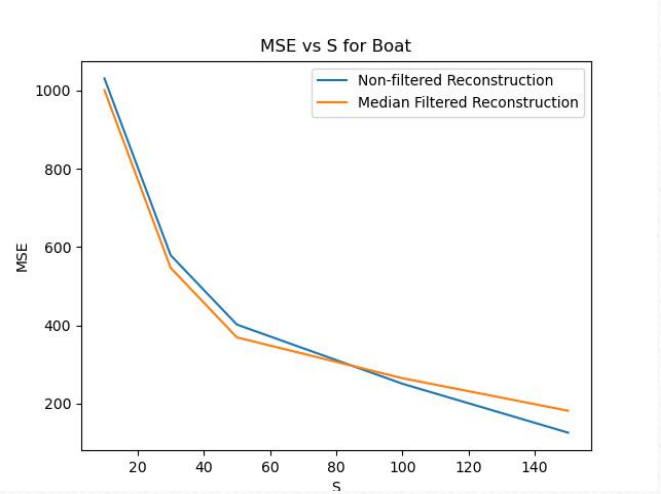
S100

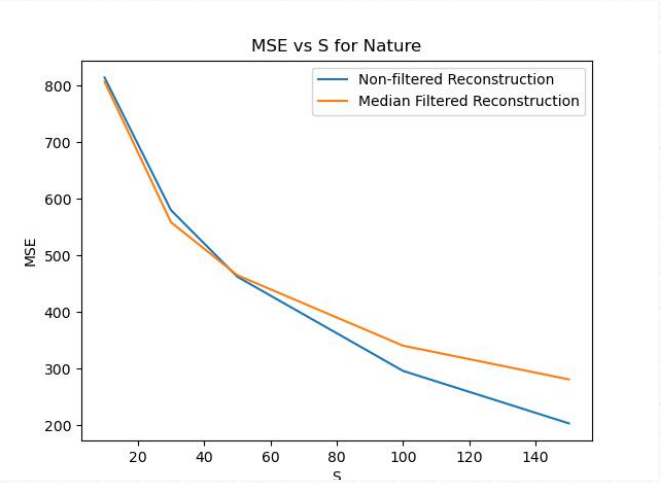


MSE vs Sensed Pixels

As the number of sensed pixels in a block decreases the amount of data points the regeression algorithmis working with also decreases. Thus the accuracy of the models produced are reduced and the MSE increases with decreasing number of sensed pixels in a block.

In the images to the right, the mean squared error greatly increases for fewer sensed pixels. Interestingly, the change in MSE between S=20 and S=10 for the boat image is considerably higher than the change in MSE between S=20 and S=10 for Nature. Whilst proportionally the Nature image had a much smaller percentage of the pixels in a block that are sensed, the image must be much more uniform over each block.

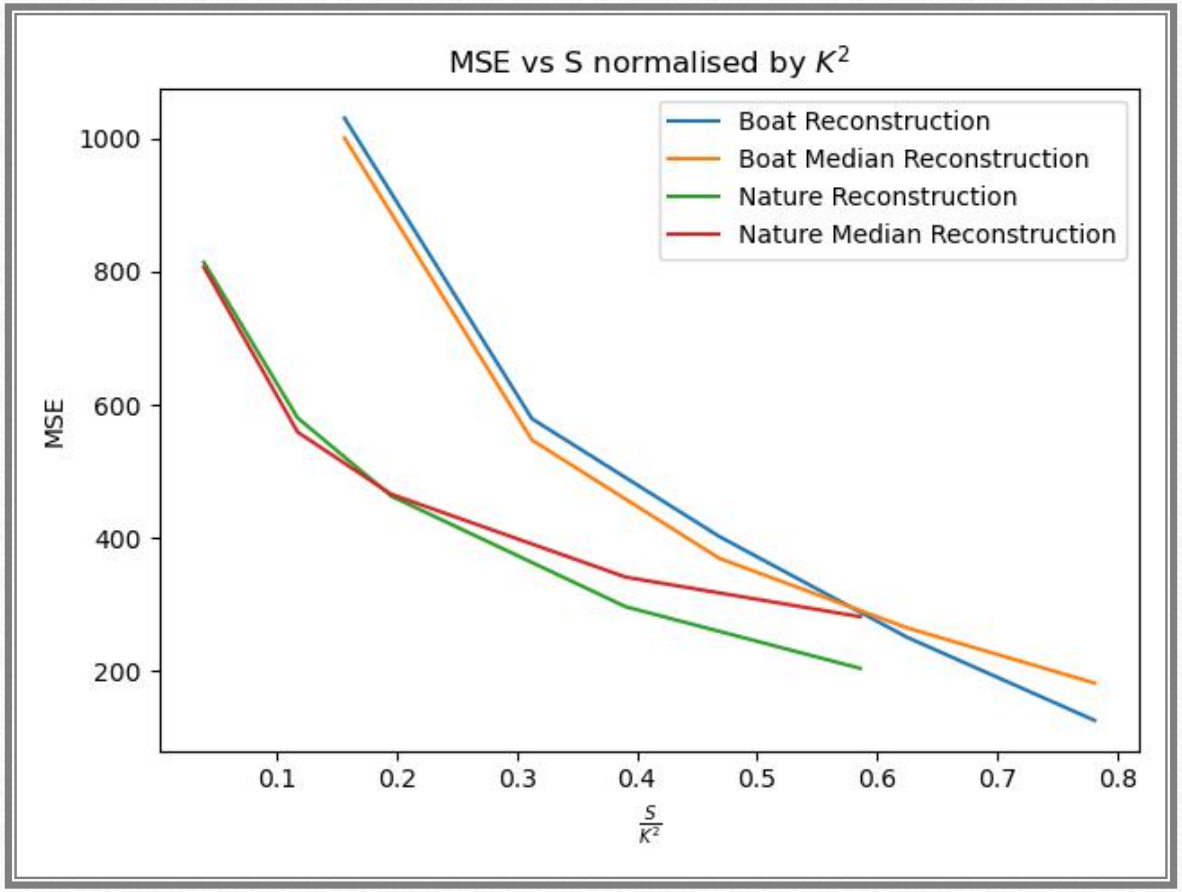




Normalised MSE vs Sensed Pixel Comparison

In the plot it is clear that median filtering has a closely related, but slightly varying impact on the MSE of an image. For the higher numbers of sensed pixels the median filter increases the MSE, but for the lower values of sensed pixels the median filter decreases the MSE.

As explained on previous slides, the median filter smoothes variation in an image: for a 3x3 filter any lines of width 1px are eliminated. Therefore, for the images with a great number of sensed pixels there is a higher likelihood that intricate details are actually a result of a detailed image rather than noise. Conversely, when the corrupted image has very few sensed pixels the mean squared error of the median filter is lower because there is minimal of the original picture's detail remaining and all of the pixel variation eliminated by the filter is due to noise in the reconstruction.



Reconstructed Boat – Median Size = 20

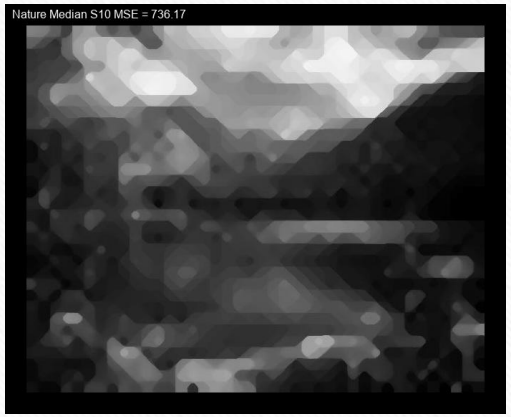
S64

Self exploration was undertaken to see what a massive increase in the size of the median filter would result in. As can be observed in the images below all of the reconstructions except S=10 have almost the exact same MSE. The median filter removed any small details any smoothed out areas of similar intensities. It is important to note, however, that it is no longer possible to observe the reconstruction blocks for each image. The image blocks are considerable smaller than the median filter size and thus any variation in blocks was removed by being considered as a noisy detail.

Reconstructed Nature – Median Size = 20

S256

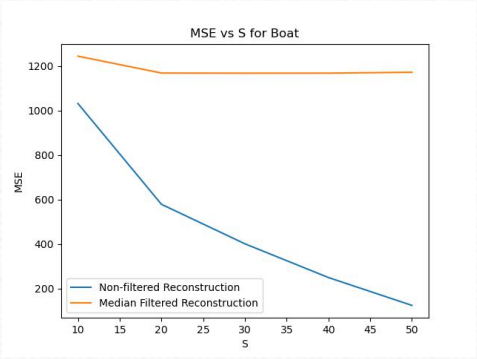
Similarly to the boat image, a large median filter reduces image detail. However, to a human observer, I believe the large median filter makes it easier to determine what the image is. The S=10 image is incredibly interesting, because the block shapes are reduced, but it produced a diagonal fractal pattern throughout.

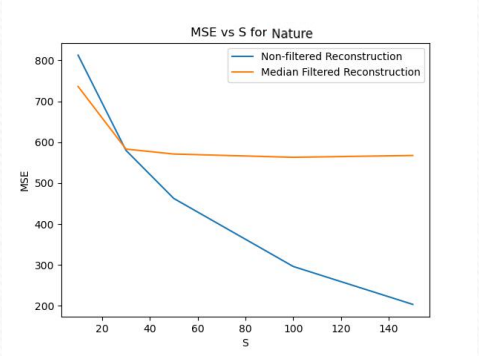


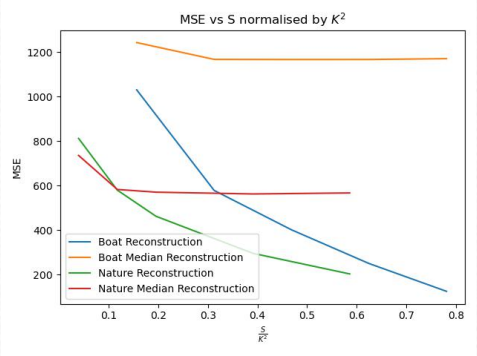
Self Motivated Explorations

With the filter size equal to 20 much of the image detail is lost. It is very surprising how much the S=10 nature image's MSE is reduced by a large median filter. This clearly shows that the rigid separation of the reconstruction into blocks is not effective when there are very few datapoints.

In general, the MSE is increased with a large median filter, but it is possible that the division between individual blocks can have a very large impact on sparse reconstructions. This is however, not the case for the boat which has an even smaller block size.







Data Analytics Shape the Future of Earth Observation
https://www.geospatialworld.net/blogs/future-of-earth-observation/

Future Plans

Based off my self exploration of the median filters it would be interesting to run a variety of median filters and determine which size of median filter is the most effective for the corresponding number of sensed pixels.

I would also like to try reconstruction of an image, median filtering and then replacement of pixels with their corresponding original pixels. It would be interesting to see the effect of this on both the MSE and to a human visual interpretation.

Collaboration

Who did you share and debate ideas with while working on this project?

Apart from during our class discussion, I did not debate any ideas with others outside class-time.

Who did you share code with while working on this project?

I did not share code with anyone during this project.

Who did you compare results with while working on this project?

I compared results with Joanna Peng at a few stages of this project.

Who did you help overcome and obstacle and vice versa while working on this project?

Joanna Peng helped me overcome and issue with what I should do with models that did not converge. I did not help others.

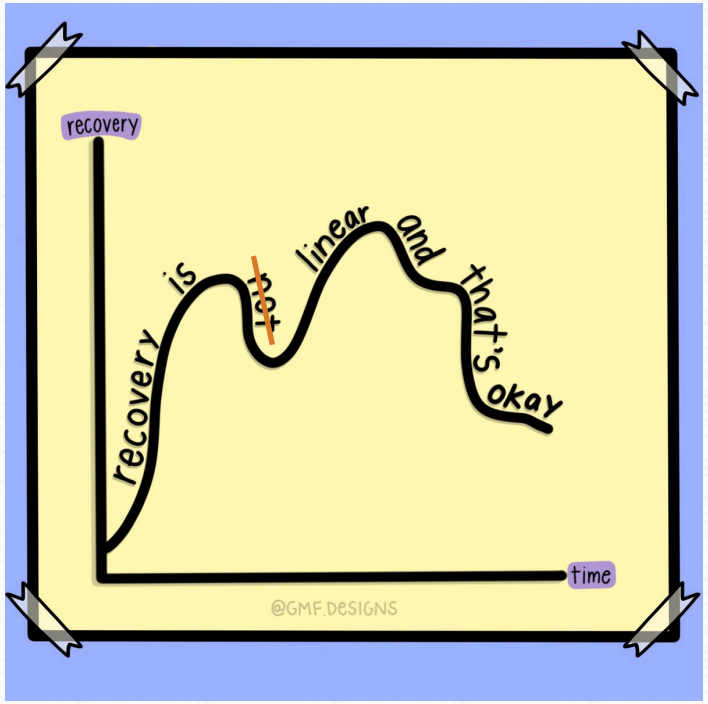
Coding Resources

- Harris, C.R. et al., 2020. Array programming with NumPy. *Nature*, 585, pp.357–362 .
- Hunter, J.D., 2007. Matplotlib: A 2D graphics environment. *Computing in science & engineering*, 9(3), pp.90–95.
- McKinney, W. & others, 2010. Data structures for statistical computing in python. In *Proceedings of the 9th Python in Science Conference*. pp. 51–56.
- Pedregosa, F. et al., 2011. Scikit-learn: Machine learning in Python. *Journal of machine learning research*, 12(Oct), pp.2825–2830.
- Umesh, P., 2012. Image Processing in Python. *CSI Communications*, 23.
- Van Rossum, G., 2020. *The Python Library Reference, release 3.8.2*, Python Software Foundation.

Resources

- Appiah, O., Martey, E.M. and Quayson, E. (2019). Effect of Window's Shape on Median Filtering. *2019 IEEE AFRICON*. doi:https://doi.org/10.1109/africon46755.2019.9133733.
- Banerjee, S., Sinha Chaudhuri, S., Mehra, R. and Misra, A. (2021). A Comprehensive Review of Median Filter, Hybrid Median Filter and Their Proposed Variants. *Advances in Smart Communication Technology and Information Processing*, pp.393–406. doi:https://doi.org/10.1007/978-981-15-9433-5_38.
- Burger, W. and Burge, M.J. (2022). Edge-Preserving Smoothing Filters. *Texts in Computer Science*, pp.497–536. doi:https://doi.org/10.1007/978-3-031-05744-1_17.
- Davies, E.R. (2012). *Computer and machine vision : theory, algorithms, practicalities*. Waltham: Elsevier.
- Davies, E.R. and Netlibrary, I. (2005). *Machine vision : theory, algorithms, practicalities*. 3rd ed. Amsterdam ; Boston: Elsevier.
- Dimililer, K. and Kavalcıoğlu, C. (2011). Gaussian Noise and Haar Wavelet Transform Image Compression on Transmission of Dermatological Images. *Advances in Computing and Communications*, pp.357–364. doi:https://doi.org/10.1007/978-3-642-22720-2_37.
- Fried, R. and George, A.C. (2011). Median Filters and Extensions. *International Encyclopedia of Statistical Science*, pp.806–806. doi:https://doi.org/10.1007/978-3-642-04898-2_361.
- Gauraha, N. (2018). Introduction to the LASSO. *Resonance*, 23(4), pp.439–464. doi:https://doi.org/10.1007/s12045-018-0635-x.
- Kleefeld, A., Breuß, M., Welk, M. and Burgeth, B. (2015). Adaptive Filters for Color Images: Median Filtering and Its Extensions. *Lecture Notes in Computer Science*, 9016, pp.149–158. doi:https://doi.org/10.1007/978-3-319-15979-9_15.
- Lederer, J. (2021). Linear Regression. *Springer Texts in Statistics*, pp.37–79. doi:https://doi.org/10.1007/978-3-030-73792-4_2.
- Marc Peter Deisenroth, A Aldo Faisal and Cheng Soon Ong (2020). *Mathematics for machine learning*. Cambridge ; New York, Ny Cambridge University Press.
- Neumaier, A. (1998). Solving Ill-Conditioned and Singular Linear Systems: A Tutorial on Regularization. *SIAM Review*, 40(3), pp.636–666. doi:https://doi.org/10.1137/s0036144597321909.
- Pitas, I. and Venetsanopoulos, A.N. (1990). Median Filters. *Nonlinear Digital Filters*, pp.63–116. doi:https://doi.org/10.1007/978-1-4757-6017-0_4.
- Sha, L. (2008). Image Compression. *Encyclopedia of GIS*, pp.472–475. doi:https://doi.org/10.1007/978-0-387-35973-1_584.
- Tan, E.-L. and Gan, W.-S. (2015). Perceptual Image Coding with Discrete Cosine Transform. *SpringerBriefs in Electrical and Computer Engineering*, pp.21–41. doi:https://doi.org/10.1007/978-981-287-543-3_3.
- Weik, M.H. (2000). Huffman coding. *Computer Science and Communications Dictionary*, pp.738–738. doi:https://doi.org/10.1007/1-4020-0613-6_8512.

Closing Thoughts



Recovery isn't Linear – The Importance of Realistic Recovery
https://youmatter.988lifeline.org/recovery-isnt-linear-the-importance-of-realistic-recovery/