# Recovering astronomical images with deep neural network supported bispectrum processing

**Jacob Lucas, Brandoch Calef**
*The Boeing Company*
**Trent Kyono**
*The Boeing Company, UCLA Computer Science*

**Abstract**

Bispectrum processing is a well-established tool for phase retrieval in speckle imaging. Recent advancements in image processing with neural networks imply great effectiveness with denoising, in-painting, and image recovery, suggesting that the application of a customized neural network to the bispectrum could improve the quality of the reconstructed phase. Motivated by this, we explore the application of deep neural networks to assist with and enhance the performance of a standard bispectrum phase retrieval algorithm.

## 1. Introduction

Bispectrum processing is a well-studied tool for phase retrieval in speckle imaging. It has been shown that the quality of the reconstruction can be limited by noise [14]. Given the stellar performance of deep learning in super-resolution and denoising, it seems likely that such methods could be successfully applied to denoise the bispectrum and reduce its sensitivity to noise.

The method described here supplies a custom deep convolutional neural network with a selection of noiseless and noisy bispectra for training. Once trained the network is applied to heretofore unseen noisy data and an image reconstructed from this 'corrected' bispectra. Here we describe the formation of the training data, the network used, and the training of said network. Also described is the reconstruction and results.

## 2. Related Works

Convolutional neural networks (CNNs) are one of the most widely accepted methods in computer vision and are status quo for image classification, segmentation, and detection tasks. With the recent success of these algorithms in imaging competitions, such as Kaggle, ImageNet, etc., coupled with the advancements in deep learning libraries, GPU hardware acceleration and data availability, these methods have received heightened interest in the astronomical and space situational awareness (SSA) domain. These recent advancements in image processing with CNNs imply great effectiveness with denoising, in-painting, and image recovery, suggesting that the application of a customized neural network to the bispectrum could improve the quality of the reconstructed phase.

Several related works have demonstrated significant improvements to state-of-the-art performance using CNNs for image denoising [10, 15, 16]. Specifically, stacked denoising autoencoders were presented as early as 2010 by [13]. Their methodology has been fine-tuned over the years with implementations, such as the *U-Net* architecture, yielding state-of-the-art in many image segmentation tasks [12]. Specifically for denoising, [5] successfully removed X-ray computed tomography (CT) artifacts using the *U-Net* architecture.

Further, [8] successfully used the *U-Net* for image restoration with state-of-the-art performance. We draw motivation from this architecture and recent success and apply it to bispectrum phase retrieval as shown in Figure 1.

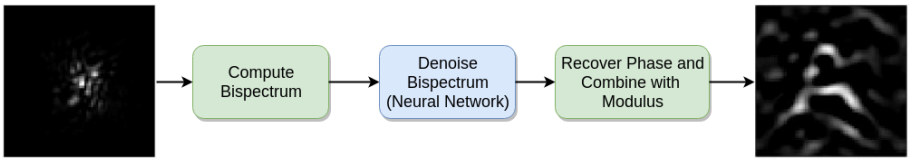


Figure 1: Neural network in bispectrum processing pipeline.

## 3. Methodology

The training data for this analysis was generated in Matlab. The MNIST [7] set of handwritten numbers was used as the base. These images are functionally very similar to astronomical images, as both have an object with distinct features roughly centered on a black background. A point spread function (*PSF*) was generated to correspond to each base image. A unique *128x128* pixel *PSF* representing light turbulence was simulated for each base image using phase screens with Kolmogorov statistics [11] with an approximate Fried parameter of 50cm, and an aperture of 3.6m. The base image set was then resized from *28x28* to *128x128* and convolved with the *PSF* image set, resulting in the reference or *noiseless* data set. The resulting image set was cropped to *64x64* to reduce bispectrum/training computation time. A copy of the noiseless data set was then created and random Gaussian noise added. This noisy data set will be referred to as the *degraded* set. Examples of these image sets are shown in Figure 2.

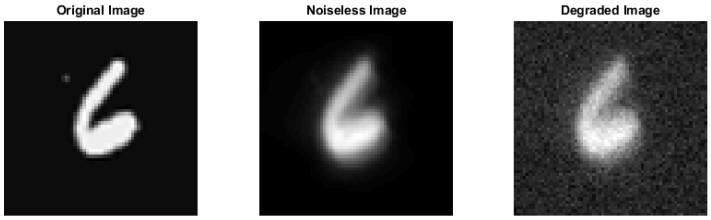


Figure 2: Examples of image sets. the *noiseless* image (center) represents the original convolved with the *PSF*, the *degraded* image (right) represent the *noiseless* with added Gaussian noise.

The approach chosen for this paper was to denoise the bispectrum, requiring that the bispectrum be calculated for each *noiseless* and *degraded* image. The bispectrum is defined as [11]:

$$B(\vec{f}_1, \vec{f}_2) = I(\vec{f}_1)I(\vec{f}_2)I^*(\vec{f}_1 + \vec{f}_2)$$

Where *I* is the *Fourier transform* of the image, $\vec{f}_1$ ranges over the full *64x64* pixel image, and $\vec{f}_2$ was given a small range of from -2 to 2 pixels to reduce computation time and the size of the result. The final size of a single training 'frame' was *64x64x5x5*. Leveraging symmetry of the bispectrum would allow this size to be greatly reduced [9], however this was not taken advantage of for this study. Cropping the image sets from *128x128* to *64x64* has a degrading effect on the reconstruction, limiting the achievable quality of the reconstructed image.

Several approaches were taken in preparing the training data. The bispectrum being a complex quantity, two ways of converting this to real values for training are to simply separate into real and imaginary parts, or to compute the phase and amplitude. Separating into real and imaginary results in fields with less than desirable statistics for training. The core of the array has values many orders of magnitude larger than those at the edges, leading to significant issues with the regression cost function. In order to prepare the data the dynamic range needed to be decreased without corrupting the bispectrum. To do this the pixel standard deviation was calculated for the data set. The training set was then divided by the standard deviation array and scaled to have a maximum range of -1 to 1. A comparison of 2-d slices of the bispectrum before and after scaling is shown in Figure 3. Using phase and amplitude was explored, however better results were seen by splitting into real and imaginary arrays. The *64x64x5x5* frames were reshaped into 25 *64x64* slices for network training.

## 4. Network Architecture and Experimental Settings

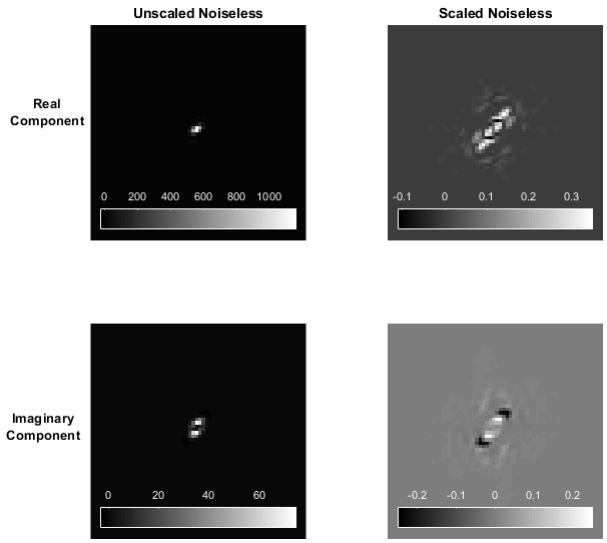


Figure 3: 2-d Slices of the *Noiseless* bispectrum before and after scaling. Plot scales are unique to better show detail.

For this experiment we employed the aforementioned *U-Net* architecture shown in Figure 4. All of the convolutional layers used a *ReLU* activation function except for the last, which used *tanh*. We initialized each kernel to Glorot normal initialization [4]. Because the bispectrum contains both a real and imaginary image, we attempted a Siamese neural network with shared parameter weighting in each convolutional layer [3]. However, performance was not improved potentially due to the magnitude of the differences between the values of the real and imaginary images. For input image augmentation, we applied random augmentation to each training image with the following specification: rotation within $\pm 25$ degrees, horizontal flips, vertical flips, and zoom within $\pm 10\%$. Our dataset consisted of 11000 image pairs, from which we used a 80% training, 10% validation, and 10% test split. We used the *Adam* optimizer with learning rate set to $1e^{-4}$. Mean squared error (*MSE*) was used as the loss function for simplicity, but a trade study may find a more effective option. The CNN was trained for 20 epochs until validation mean squared error did not improve, and the best validation mean squared error was saved for inference/prediction. Image augmentation was not used during inference.

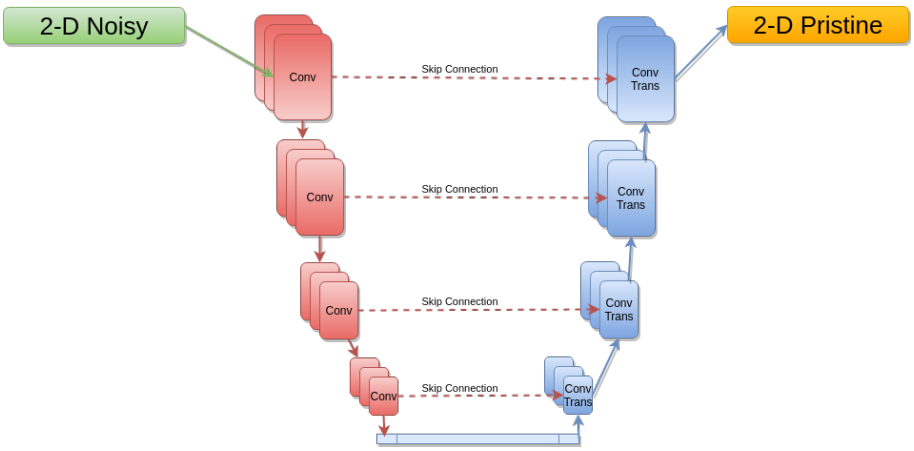


Figure 4: U-Net architecture used in denoising 2-D images. Last output layer used *tanh* activation, all other layers were *ReLU*.

For reproduction, all neural networks were trained using Python 3 and *Keras* in conjunction with TensorFlow [2, 1]. Operating system and hardware specifications include Ubuntu Linux 16.04 with dual Nvidia 1080 Ti GPUs. Because the goal of this work was a feasibility investigation, we did not tune or search for optimal hyperparameters.

Following the denoising of the bispectra by the network, the network prediction is rescaled (i.e. the pre-training standard deviation scaling is reversed), the real and imaginary components are combined, and a recursive reconstructor [11] is used to recover the object phase of a 50 frame ensemble. The *Fourier* modulus was then obtained via the *Labeyrie* method [6], and the two combined to form the final image. In Figure 7 and Figure 8 this result is referred to as the *network reconstruction*.

The *combined reconstuction* shown in Figure 7 and Figure 8 was obtained by combining the magnitude of the rescaled *network corrected* data with the phase of the *degraded* data set and applying the recursion. This has the advantage of retaining some of the noise suppression from the network, in addition to much of the image information that the network tended to corrupt.

## 5. Results

The images below show the improvement gained by application of the network to the the noisy data. The quality of the *noiseless* reconstruction is limited due to lack of diversity of the *PSF* over the simulated collection time. The network prediction shows clear reduction in noise, however in the process of denoising it has changed the content significantly enough to prevent an accurate reconstruction. Figure 6 shows slices of the bispectrum before and after application of the network. The network noise reduction is obvious, however closely comparing the *noiseless* and *network predicted* images reveals subtle differences. Figure 5 shows the residuals found by subtracting the *corrected* data from both the *noiseless* and *degraded*.
These differences lead to the degradation in the reconstruction shown in Figure 7. Again the noise present in the *degraded* data has been suppressed, however there is a significant reduction in the quality of the reconstructed image. Some improvement in noise reduction without degradation of the image were found using the *combined reconstruction* described in the previous section. In all cases this combination resulted in reduced noise, and in some cases it improved the visual quality of the reconstruction.

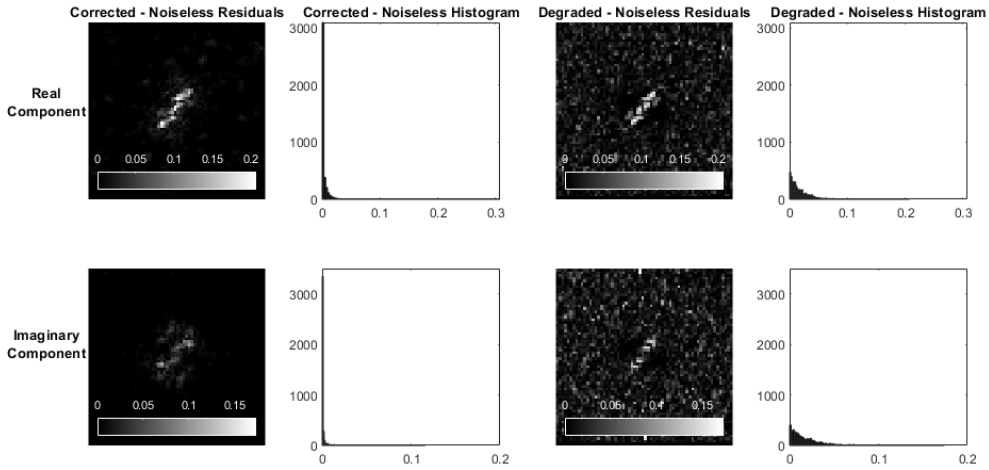


Figure 5:
Residuals for an arbitrary 2-d bispectrum slice found by subtracting the *noiseless* data from both the *corrected* data (left) and *degraded* data (right).

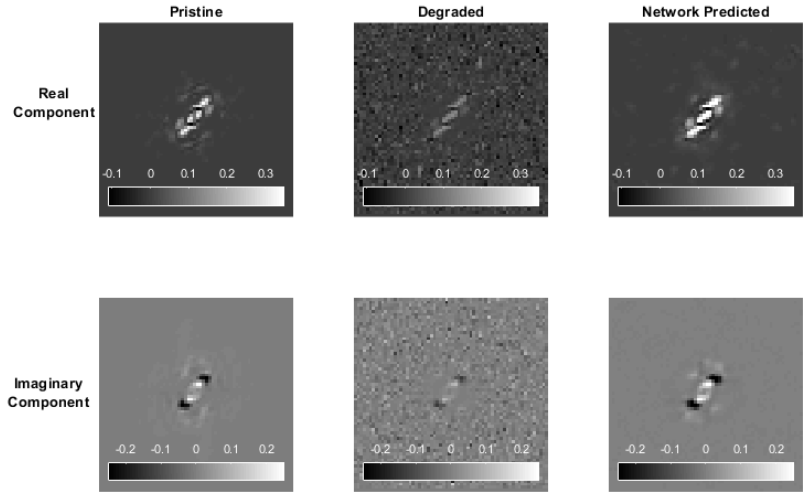


Figure 6:
Network predictions as seen on 2-d slices of the bispectrum. While the network prediction is a clear improvement, some differences remain. All real and imaginary component plots have the same respective scales.

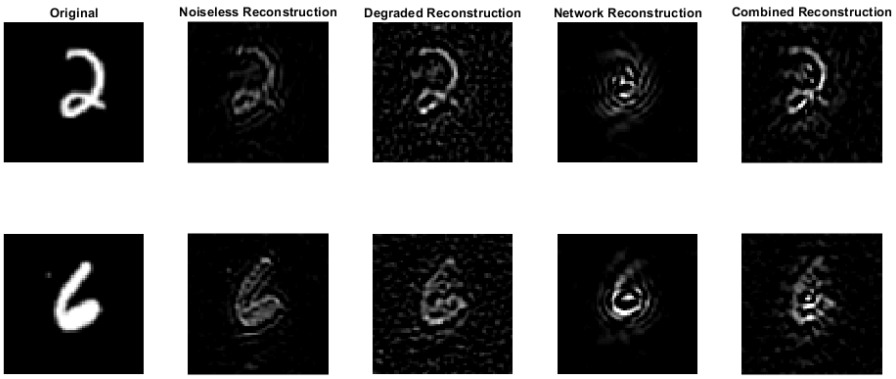


Figure 7: Network predictions after reconstruction with a triangle filter applied. The network reconstruction on its own shows some degradation, however the combined reconstruction is on par with the degraded with less noise. The *Original* column represents the images prior to convolution with *PSF*. Scales are unique to each plot to better show detail.

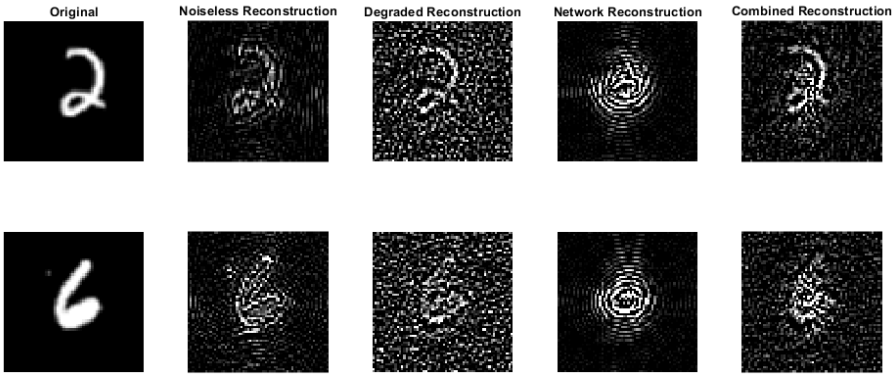


Figure 8: Network predictions after reconstruction with no filter. As in the filtered plots, the combined has less noise than the degraded. Scales are unique to each plot to better show detail.

## 6. Conclusion

This effort indicates that applying a deep convolutional neural network to noisy bispectra can reduce the noise in the reconstructed phase, and a combination of *degraded* and *network corrected* can improve the quality of the reconstruction even if the *network corrected* is less than ideal. While the improvements shown here are minimal, the training set used was rather small. Increasing the size of the training set and expanding the augmentation parameters are standard practice for improving network accuracy. A code base has been constructed to enable rapid production of unique *PSF*s to allow for straightforward expansion of the training set, and a network architecture proven to denoise has been established. Further work will involve a more extensive data set as well as a further look into the feasibility of denoising the bispectrum phase and amplitude.

## References


[1] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[2] François Chollet et al. Keras, 2015.

[3] S. Chopra, R. Hadsell, and Y. LeCun. Learning a similarity metric discriminatively, with application to face verification. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 539–546 vol. 1, June 2005.

[4] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In Yee Whye Teh and Mike Titterington, editors, *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pages 249–256, Chia Laguna Resort, Sardinia, Italy, 13–15 May 2010. PMLR.

[5] Yoseob Han and Jong Chul Ye. Framing u-net via deep convolutional framelets: Application to sparse-view CT. *CoRR*, abs/1708.08333, 2017.

[6] A. Labeyrie. Attainment of Diffraction Limited Resolution in Large Telescopes by Fourier Analysing Speckle Patterns in Star Images. , 6:85, May 1970.

[7] Yann LeCun and Corinna Cortes. MNIST handwritten digit database. 2010.

[8] Xiao-Jiao Mao, Chunhua Shen, and Yu-Bin Yang. Image restoration using convolutional auto-encoders with symmetric skip connections. *CoRR*, abs/1606.08921, 2016.

[9] P. Negrete-Regagnon. Practical aspects of image recovery by means of the bispectrum. *Journal of the Optical Society of America A*, 13:1557–1576, July 1996.

[10] Tal Remez, Or Litany, Raja Giryes, and Alexander M. Bronstein. Deep convolutional denoising of low-light images. *CoRR*, abs/1701.01687, 2017.

[11] Roggeman and Welsh. *Imaging Through Turbulence*. CRC Press, 1996.

[12] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. *CoRR*, abs/1505.04597, 2015.

[13] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine Manzagol. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *J. Mach. Learn. Res.*, 11:3371–3408, December 2010.

[14] Tengfei Wu, Ori Katz, Xiaopeng Shao, and Sylvain Gigan. Single-shot diffraction-limited imaging through scattering layers via bispectrum analysis. *Opt. Lett.*, 41(21):5003–5006, Nov 2016.

[15] Junyuan Xie, Linli Xu, and Enhong Chen. Image denoising and inpainting with deep neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 341–349. Curran Associates, Inc., 2012.

[16] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *IEEE Transactions on Image Processing*, 26(7):3142–3155, July 2017.